

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平 6-110684

(43) 公開日 平成 6 年 (1994) 4 月 22 日

(51) Int. Cl. ⁵

識別記号

庁内整理番号

F I

技術表示箇所

G 0 6 F 9/38

3 3 0 B 9193-5 B

3 7 0 X 9193-5 B

審査請求 有 請求項の数 12

(全 35 頁)

(21) 出願番号 特願平 5-162902

(22) 出願日 平成 5 年 (1993) 6 月 30 日

(31) 優先権主張番号 928851

(32) 優先日 1992 年 8 月 11 日

(33) 優先権主張国 米国 (US)

(71) 出願人 390009531

インターナショナル・ビジネス・マシーンズ・コーポレーション

INTERNATIONAL BUSINESS MACHINES CORPORATION

アメリカ合衆国 10504、ニューヨーク州
アーモンク (番地なし)

(72) 発明者 フィリップ ジョージ エマ

アメリカ合衆国 06811 コネチカット州
ダンペリー フォックス デン ロード
28

(74) 復代理人 弁理士 谷 義一 (外 3 名)

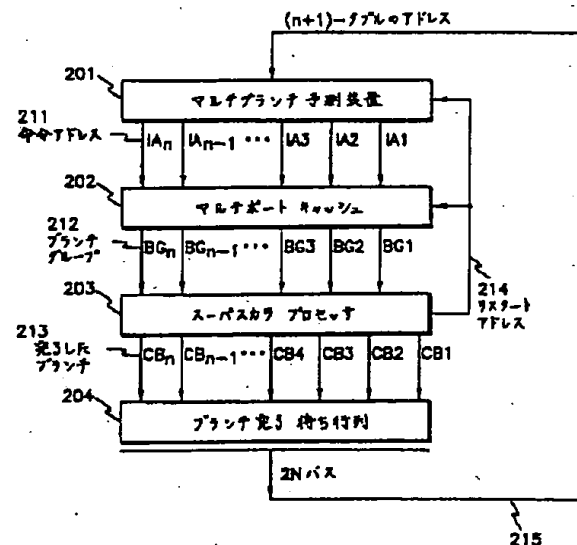
最終頁に続く

(54) 【発明の名称】 スーパスカラ処理システムおよび方法

(57) 【要約】

【目的】 同時に予測を行う。

【構成】 前のブランチ命令のブランチアドレスからストアアドレスを生成し、マルチブランチヒストリテーブルのエントリであって、前のブランチ命令に対応するエントリであり、しかも、ブランチアドレスを含むとともに、複数のブランチ命令のそれぞれの前の目標アドレスを含むエントリを前記ストアアドレスを用いてストアし、エントリをマルチブランチヒストリテーブルから検索し複数のブランチ命令のそれぞれに対して、前記エントリの前の目標アドレスを前記予測された未来目標アドレスとして適用する。



【特許請求の範囲】

【請求項1】 複数のブランチ命令に対してそれぞれ未来目標アドレスを同時に予測する方法であり、前記複数のブランチ命令が前の実行で前のブランチ命令に順次従った方法であって、

前記前のブランチ命令のブランチアドレスからストアアドレスを生成するステップと、
マルチブランチヒストリテーブルのエントリであって、前記前のブランチ命令に対応するエントリであり、しかも、ブランチアドレスを含むとともに、前記各ブランチ命令の前の目標アドレスを含むエントリを前記ストアアドレスを用いてストアするステップと、
前記エントリを前記マルチブランチヒストリテーブルから検索するステップと、

前記各ブランチ命令に対して、前記エントリの前の目標アドレスを前記予測された未来目標アドレスとして適用するステップとを備えたことを特徴とする方法。

【請求項2】 スーパスカラプロセッサの命令キャッシュに結合されたシステムであり、複数のブランチ命令を同時に予測し、前記予測を用いて、前記命令キャッシュに置かれる命令のブランチグループを取り出すシステムであって、

(a) 第1の複数のブランチ命令の目標アドレスを、前記第1の複数のブランチ命令の実行履歴に基づき同時に予測する1次マルチブランチヒストリテーブル手段と、

(b) 該1次マルチブランチヒストリテーブル手段に対して少なくとも1ブランチ命令だけ遅れ、第2の複数のブランチ命令の目標アドレスを、前記第2の複数のブランチ命令の実行履歴に基づき同時に予測し、前記1次マルチブランチヒストリテーブル手段による予測に際してダイバーゼンスを検知するために用いられる2次マルチブランチヒストリテーブル手段と、

(c) 該2次マルチブランチヒストリテーブル手段と前記1次マルチブランチヒストリテーブル手段とに結合され、該1次マルチブランチヒストリテーブル手段による予測と、前記2次マルチブランチヒストリテーブル手段による予測とを比較し、ダイバーゼンスを検知する比較手段と、

(d) 前記1次マルチブランチヒストリテーブル手段と前記2次マルチブランチヒストリテーブル手段とに結合され、前記1次マルチブランチヒストリテーブル手段と前記2次マルチブランチヒストリテーブル手段の前記実行履歴を、スーパスカラプロセッサにより実行されたブランチ命令の系列の結果に基づき更新する更新手段とを備えたことを特徴とするシステム。

【請求項3】 請求項2に記載のシステムにおいて、前記1次および第2マルチブランチヒストリテーブル手段は、それぞれ、マルチブランチヒストリテーブルを備え、該マルチブランチヒストリテーブルのエントリは、

前に実行されたブランチ命令に対応し、前記前に実行されたブランチ命令の次の複数のブランチ命令のブランチおよび目標アドレスを前の実行から指定することを特徴とするシステム。

【請求項4】 請求項2に記載のシステムにおいて、前記更新手段は、

(e) 前記1次マルチブランチ予測手段による予測を表すアドレスを保持する1次保留予想待ち行列と、

(f) 前記2次マルチブランチ予測手段による予測を表すアドレスを保持する2次保留予想待ち行列と、

(g) 前記スーパスカラにより実際に実行されるブランチを表すアドレスを保持する更新待ち行列と、

(h) 該更新待ち行列に保持された前記アドレスと、前記1次および2次保留予想待ち行列に保持された予測を表す前記アドレスとを比較し、前記1次および2次ブランチヒストリテーブルにそれぞれ対する不必要な更新をフィルタアウトする第2比較手段とを備えたことを特徴とするシステム。

【請求項5】 複数のブランチ命令を同時に予測し、多重性の高い命令をデコードし実行するプロセッサより用いるため、その予測を用いて複数の非結合命令系列を同時に取り出すシステムであって、

(a) 前記プロセッサに結合され、少なくとも2つのブランチを行う命令の系列を記述するレコードを受信する更新待ち行列であって、前記レコードは少なくとも3つのアドレスを有し、前記複数のアドレスは前記複数のブランチを行う命令により接続された複数の非結合命令系列を識別し、前記複数のアドレスは第1部分および第2部分を含み、前記第1部分は開始アドレスであり、前記第2アドレスは残りの複数のアドレスであり、前記第1部分は第1命令系列を識別し、前記第2部分は前記複数の非結合系列の残りの複数の命令系列を識別する更新待ち行列と、

(b) 前記更新待ち行列に結合されたマルチブランチヒストリテーブルであって、前記更新待ち行列からの前記レコードを受信し、しかも、前記レコードの前記第2部分を、将来使用するため、前記マルチブランチヒストリテーブルの場所であって、前記レコードの前記第1の部分により決定された場所にストアするマルチブランチヒストリテーブルと、

(c) 前記プロセッサと前記マルチブランチヒストリテーブルに結合され、前記プロセッサからの命令アドレスを受信する事前取り出しアドレスレジスタであり、前記レコードの前記前にストアされた第2部分の前記マルチブランチヒストリテーブルからの読み出しを、前記命令アドレスに対応してイネーブルにする事前取り出しアドレスレジスタであって、前記前にストアされた前記第2部分が前記プロセッサにより受信される事前取り出しアドレスレジスタと、

(d) 前記マルチブランチヒストリテーブルと前記事前取

り出しアドレスレジスタに結合された保留予想待ち行列であって、前記事前取り出しアドレスレジスタからの前記命令アドレスを、前記マルチブランチヒストリテーブルからの前記読み出しを用いて受信しストアする保留予想待ち行列であり、前記命令アドレスは前記読み出しを有するとともに、前記プロセッサにより前に実行された複数の非結合命令系列を識別する複数のアドレスよりなり、前記複数のヒストリのレコードである保留予想待ち行列と、

(e) 前記保留予想待ち行列と前記更新待ち行列とに結合され、前記保留予想待ち行列からの前記ヒストリのレコードと、前記更新待ち行列により受信された前記レコードとを比較する比較器であって、前記保留予想待ち行列からの前記ヒストリのレコードと、前記更新待ち行列により受信された前記レコードとが一致しない場合にのみ、前記更新待ち行列により受信された前記レコードを前記更新待ち行列にストアすることができるシグナルを供給する比較器とを備えたことを特徴とするシステム。

【請求項6】 請求項5に記載のシステムにおいて、前記プロセッサからの前記命令アドレスと前記マルチブランチヒストリテーブルからの前記読み出しの部分を受信するセレクト手段であって、前記プロセッサからの前記命令アドレスか、あるいは、前記読み出しの前記部分か、のいずれかを選択し、しかも、前記選択を前記事前取り出しアドレスレジスタに経路指定するセレクト手段をさらに備えたことを特徴とするシステム。

【請求項7】 複数のブランチ命令を同時に予測し、多重性の高い命令をデコードし実行するプロセッサにより用いるためその予測を用いて複数の非結合命令系列を同時に取り出すシステムであって、

第1の複数のブランチを予測するための1次マルチブランチヒストリテーブルと、

第2の複数のブランチを予測するための2次マルチブランチヒストリテーブルであって、前記第2の複数のブランチが前記第1の複数のブランチからの少なくとも1つのブランチ命令によりオフセットされる2次マルチブランチヒストリテーブルと、

前記第1の複数の予測されたブランチが正しいか否かを判定する手段と、

前記第1の複数の予測されたブランチが正しい場合、前記第1の複数の予測されたブランチを選択する手段とを備えたことを特徴とするシステム。

【請求項8】 請求項7に記載のシステムにおいて、前記1次マルチブランチヒストリテーブルは、

第1の1次ブランチグループと第2の1次ブランチグループをそれぞれ含むエン트리と、

前記1次ブランチグループを、前記第1の1次ブランチグループからの目標アドレスか、あるいは、前記1次マルチブランチヒストリテーブルからの前記第2の1次ブランチグループからの目標アドレスに基づき、選択する

手段とを備えたことを特徴とするシステム。

【請求項9】 請求項7に記載のシステムにおいて、前記判定手段は、前記1次マルチブランチヒストリテーブルからの前記選択されたエントリの第2の1次ブランチグループと、前記2次マルチブランチヒストリテーブルからの前記選択されたエントリの第1の2次ブランチグループとを比較する手段を備えたことを特徴とするシステム。

【請求項10】 請求項7に記載のシステムにおいて、前記選択手段は、

前記1次マルチブランチヒストリテーブルからの前記選択されたエントリの前記第2の1次ブランチグループと、前記第2のマルチブランチヒストリテーブルからの前記選択されたエントリの前記第1の2次ブランチグループとを比較した結果、等しい場合、前記選択されたエントリを、前記1次マルチブランチヒストリテーブルから選択する手段と、

前記1次マルチブランチヒストリテーブルからの前記選択されたエントリの前記第2の1次ブランチグループと、前記第2のマルチブランチヒストリテーブルからの前記選択されたエントリの前記第1の2次ブランチグループとを比較した結果、等しくない場合、前記選択されたエントリを、前記2次マルチブランチヒストリテーブルから選択する手段とを備えたことを特徴とするシステム。

【請求項11】 複数のブランチ命令を同時に予測し、多重性の高い命令を同時にデコードし実行するプロセッサにより用いるため、前記複数の予測を用いて、複数の非結合命令系列を取り出す方法であって、

(1) 前記プロセッサに結合された更新待ち行列の少なくとも2つのブランチを行う命令を記述する記録を受信するステップであって、前記レコードは少なくとも3つのアドレスを備え、前記複数のアドレスは前記複数のブランチを行う命令により接続される複数の非結合命令系列を識別し、前記複数のアドレスは第1部分と第2部分を含み、前記第1部分は開始アドレスであり、前記第2部分は残りの複数のアドレスであり、前記第1部分は第1命令待ち行列を識別し、前記第2部分は前記複数の非結合命令待ち行列のうちの複数の残りの命令待ち行列を識別するステップと、

(2) 前記更新待ち行列に結合され、マルチブランチヒストリテーブルの前記更新待ち行列からの前記レコードを受信し、将来使用するため、前記レコードの前記部分を、前記マルチブランチヒストリテーブルの場所であって、前記レコードの前記第1部分により決定された場所にストアするステップと、

(3) 前記プロセッサからの命令アドレスを事前アドレスレジスタで受信するステップであって、前記レコードの前記前にストアされた第2部分の前記マルチブランチヒストリテーブルからの読み出しをイネーブルにし、前記

前にストアされた第2部分が前記プロセッサにより受信されるステップと、

(4) 前記事前取り出しアドレスレジスタからの前記命令アドレスを、前記マルチブランチヒストリテーブルからの前記読み出しとともに保留予想待ち行列にストアするステップであって、前記命令アドレスは、前記読み出しを有するとともに、前記プロセッサにより前に実行された複数の非結合命令系列を識別する複数のアドレスよりなるステップと、

(5) 前記ヒストリのレコードと前記更新待ち行列により受信された前記レコードとを比較器で比較し、しかも、前期保留予想待ち行列からの前記ヒストリのレコードと、前記更新待ち行列により受信された前記レコードとが一致しない場合にのみ、前記待ち行列により受信された前記レコードを前記更新待ち行列にストアすることができるシグナルを供給するステップとを備えたことを特徴とする方法。

【請求項12】 請求項11に記載の方法において、

(6) 前記プロセッサからの前記命令アドレスと、前記マルチブランチヒストリテーブルからの前記読み出しの部分とを受信するステップと、

(7) 前記プロセッサからの前記命令アドレスか、あるいは前記読み出しの前記部分のいずれかを選択するステップと、

(8) 前記選択を前記事前取り出しアドレスレジスタに経路指定するステップとをさらに備えたことを特徴とする方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、一連のブランチ結果を動的かつ同時に予測するシステムおよび方法に関する。特に、本発明は、一連の複数非結合命令セグメントを予測するシステムおよび方法であって、しかも、複数命令を並列にデコードし実行することができるスーパースカラプロセッサに命令系列を供給する並列記憶システムに対し、命令取り出しアドレスの対応する多重性を生成するシステムおよび方法に関する。

【0002】 なお、本明細書の記述は本件出願の優先権の基礎たる米国特許出願第07/928,851号の明細書の記載に基づくものであって、当該米国特許出願の番号を参照することによって当該米国特許出願の明細書の記載内容が本明細書の一部分を構成するものとする。

【0003】

【従来の技術】 適正な命令フローでは、命令は順次デコードされる。すなわち、命令は記憶装置の順次記憶場所から取り出されデコードされる。ブランチ命令はこの命令フローを混乱させる命令である。すなわち、ブランチにより、順次パス上でのデコーディングが中止され、記憶装置の新しい記憶場所から再開される。記憶装置の新しい記憶場所はブランチ目標アドレスと呼ばれる。命令

呼び出しが他の命令のデコーディングと実行とにオーバーラップするので、ブランチによりパイプラインパフォーマンスが低下する。

【0004】 例えば、ステージが、i) 命令デコード (DEC) で示す)、ii) オペランドアドレスに対するアドレス生成、ブランチ命令の場合は、ブランチ目標アドレスに対するアドレス生成、iii) オペランドを取り出すためのキャッシュアクセス (CACHEで示す)、ブランチ命令の場合は、ブランチ目標命令を取り出すためのキャッシュアクセス、iv) 入力オペランドに対する機能オペレーションを、命令による指定通りに行う実行 (EXECで示す) であるパイプラインを考察する。図7はこのパイプラインの命令フローの一例を示す。

【0005】 図7に示すように、新しい命令を各サイクルでデコードして、4つの命令を同時にオペレーションの幾つかのフェーズにすることができる。例えば、サイクル#4では、命令I1-I4は全てパイプラインにより処理される。ブランチ命令がサイクル#5でデコードされたとき、デコーディングが停止される。ブランチ目標アドレスをAGENにより計算しなければならない。しかも、目標命令がパイプラインに入る前に、(TARGETで示す) 目標命令CACHから取り出さなければならない。

【0006】 この例では、ブランチ命令 (BR) により、2つの空のスロットがパイプラインに生じる。これらのスロットは図7のEXECフェーズのサイクル9およびサイクル10の間に現れる。これらのサイクルでは、プロセッサにより有用なワークが行われておらず、よって、これらのサイクルでは、パフォーマンスが低下する。

【0007】 この例で用いた特定のブランチ命令は、最も重大でないブランチ命令である。このブランチ命令は無条件ブランチと呼ばれる。というのは、このブランチ命令は、無条件で、ブランチ命令 (BR) から目標命令 (TARGET) に転送されるからである。すなわち、ブランチ命令がデコードされたとき、制御がTARGETに転送されることは公知である。パフォーマンスのコストがかかるブランチ命令は、条件付き命令である。この命令は、ある条件が成立した場合にのみ、TARGETへの制御転送を指定する。この条件は、通常、そのブランチの前の幾つかの命令の結果により判定される。

【0008】 上述した例では、条件付きブランチ命令により、サイクルという適正なペナルティがさらに1つ課されることになるであろう。というのは、条件付きブランチは、制御がTARGETに転送されるか否かを判定するため、実行 (EXEC) を完了しなければならない。制御がTARGETに転送されないとEXECにより判定された場合、ブランチ後にデコードされた命令はそのブランチ後の (アドレスが) 次の順次命令である。従って、命令付きブランチ命令が取られないときでさえ、この例の

ブランチに関連して3サイクルだけ適正に遅延する。

【0009】条件付きブランチ命令が取られないことをデコードタイムで判定することができる場合、そのブランチ命令に関してペナルティはないことに注意すべきである。すなわち、次の順次命令を、そのブランチ命令のデコード直後にデコードすることができることに注意すべきである。しかし、そのブランチ命令がとられたことがデコードタイムに判定された場合、2つのサイクルペナルティがそのブランチに関連する。すなわち、目標アドレスを生成しなければならない。そして、目標命令を取り出さなければならない。しかし、ブランチが実行されるサイクルは省かれる。

【0010】条件付きブランチの結果をデコードタイムで予測しようとする機構は、「デコードタイム」予測機構と呼ばれる。このような機構が正確に予測する場合は、行わ(taken)れないブランチに対して3つのサイクルが省かれ、上述したパイプラインで行われたブランチに対して1つのサイクルが省かれる。1つの特定の型のデコードタイムの予測は、米国特許第 4,477,872号(発明者: Losq 他)と米国特許第 4,430,706号(発明者: Sand)に記述されたように、「デコードヒストリテーブル」と呼ばれる。

【0011】DHTはエントリのテーブルである。ここで、ブランチ命令のアドレスを構成する複数ビットに対する変換(ハッシュまたは切り捨て)に基づき、エントリにアクセスされる。エントリ自身は単一のビットを備えている。最後にブランチ命令が実行されたとき、対応するブランチ命令がとられた場合、そのビットはセットされ、他方、対応するブランチ命令がとられない場合、そのビットはセットされない。条件付きブランチ命令がデコードされたとき、ブランチ命令のアドレスを用いてDHTにアクセスされる。DHTエントリがセットされた場合、ブランチがとられたことが予想(guess)され、目標アドレスが生成され、そして、目標命令が取り出され、そのブランチ命令がデコードされた後の第3サイクルでデコードされる(よって、1サイクルだけ遅延して保管される)。DHTエントリがセットされない場合は、そのブランチが行われないことが予想される。次の順次命令はブランチ命令のデコード後のサイクルでデコードされる。DHTが誤って予測されたことが分かる

(すなわち、その予測と、EXECで計算されたブランチ結果とが一致しない)場合、対応するエントリは訂正される。従って、デコードタイム予測機構は、行われないブランチと関係する全てペナルティを回避する機会を提供し、同様に、行われたブランチに対する実行タイムペナルティ(典型的には、1サイクル)を回避する機会が提供される。ブランチペナルティの減少が、用いられた特定デコードタイム機構の精度に依存する。しかし、100%の精度を有するデコードタイム機構でも、ブランチペナルティを全て除去することはできない。

【0012】特に、ブランチが行われたときはいつでも、ペナルティは目標アドレスを生成し、目標命令を取り出す時間に等しくなる。そのため、ブランチペナルティを減少させる唯一の方法は、ブランチ命令が実際にデコードと遭遇する前に、とられたブランチを予想(anticipate)し、目標命令を取り出すことである。このことを行おうとする機構は、「事前取り出しタイム予測機構」と呼ばれる。

【0013】上述したことで明らかであるが、自治権のある事前命令取り出しエンジンが存在するものと仮定している。事前取り出しタイム予測機構が存在しない場合は、単一の事前取り出しエンジンは、i)順次命令アドレスをステップスルー(step through)するのに用いる単一のインクリメントと、ii)命令デコードによりデコードされる順次命令を保持する命令バッファと、iii)インクリメントにより生成される順次アドレスを用いて命令の順次ブロックをキャッシュから取り出し、その順次ブロックを命令バッファに置く手段と、iv)取り出されたブランチ命令の場合に、新しい開始アドレス(例えば、ブランチ目標アドレス)をインクリメントに供給するプロセッサのための手段とを備えることができる。

【0014】「自治権がある」ということは、事前取り出しエンジンが自由走行し、命令デコードから比較的独立していることを意味する。よって、行われたブランチがない場合は、命令バッファは、命令デコードによりデコードされる次の順次命令を常に含むことになり、正確に予想(guess)された条件付きブランチであって、行われないブランチに対してペナルティはないことになる。

【0015】事前取り出しタイム予測機構は、(デコードタイム機構とは対象的なものであり、デコードに関連してオペレートする)事前取り出しエンジンに組み込まれた機構である。事前取り出し機構は、自治権を有したまま、行われることが予測されたブランチの事前取り出しを行った直後に、事前取り出し命令をブランチ目標バスにリダイレクトしなければならない。そうすることにより、事前取り出しタイム機構は、ブランチ命令がデコードされたときに、命令バッファがブランチ目標命令を含むことを保証する。この試みが成功した場合、ブランチ目標命令をデコード命令またはブランチ命令直後にデコードすることができる。従って、事前取り出しタイム機構は、正確に予測したとき、行われたブランチに対してさえ、全てのブランチペナルティを除去する。

【0016】全てではないが、大部分の事前取り出しタイム予測機構は、米国特許第 3,559,183号に最初に記述された「ブランチヒストリテーブル」(発明者: Sussenguth)の対する変形である。これは、上述したデコードヒストリテーブルの事前タイムに類似するものである。すなわち、BHTはエントリのテーブルである。ここで、事前取り出しが行われている命令のブロックのアドレスを構成するビットに対する変換(ハッシュまたは切

り捨て)に基づきエントリにアクセスされる。

【0017】次の3つの理由により、エントリ自身はDHTエントリよりはるかに複雑である。第1の理由としては、BHT機構は事前取り出しタイムにて「盲滅法」でオペレートする。すなわち、これらのブロックの内容を試験することができるという利点もなく、すなわち、命令デコーダの支援がなく命令のブロックを取り出すだけだからである。第2の理由は、非常に重要なことであるが、取り出しクォンタムが命令境界に必ずしも位置合わせしていない命令のブロックである。そのため、複数命令と命令フラグメントを含むことができる。取り出しクォンタムは単一の命令ではない。第3の理由は、BHT機構はブロック内に行われたブランチが存在することを予測したときは、BHT機構は予測目標アドレスを出しななければならない。

【0018】従って、BHTエントリは、命令の関連するブロックが行われたブランチを含むことを、プロセッサがそのブロック内に行われたブランチに以前に遭遇したことに基づき、識別することができなければならない。さらに、BHTエントリは、ブロック内のどこに値行われたブランチ命令が駐在するかを識別することができなければならない。というのは、ブロックが入力された場所により、すなわち、現ブランチアクティビティにより、特定のブランチ命令が現命令取り出しに関係することができるか、あるいは関係することができないからである。最後に、ブランチ目標アドレスを指定し、その結果、特定のブランチが現事前取り出しアクティビティに関係する場合、事前取り出しを直ちに目標バスにリダイレクトすることできなければならない。

【0019】ブランチを行うブランチ命令にプロセッサが遭遇したとき、そのプロセッサはブランチのアドレスに基づきBHTエントリを作成する。エントリ自身はブランチ目標アドレスを含むことになる。そのブランチを含むコードの特定のセクションに再び遭遇した場合、ブランチ命令が事前取り出しされたとき、事前取り出しは、BHTエントリにより、ヒストリの目標アドレスにリダイレクトすることができる。

【0020】BHTが事前取り出しをリダイレクトしたとき、BHTはプロセッサでのこのアクションに関する情報(例えば、BHTが行われたブランチの存在を「確信」するアドレスと、ブランチのアドレス)を待ち行列化する。事前取り出しされたコードをプロセッサが後で実行するように、BHTが正しいか否かを判定する機会がプロセッサには3回ある。BHTがそのブランチを正確に予想する場合、そのブランチに関係するペナルティは存在しないが、不正確な予測を行ったことに対して重大なペナルティを関係させることができる。(「予想(guess)」という用語と「予測」という用語は置換可能であり、その分野では同義語である。)プロセッサパイプラインには、BWG(branch-wrong-guess)を検知するこ

とができる3つの時点が存在する。これらの時点とは、デコードタイム(DEC)と、アドレス生成タイム(AGEN)と、実行タイム(EXEC)とに存在する。

【0021】デコードタイム(DEC)では、BWGは2つの方法のうちの1つで明らかにすることができる。第1に、デコーダが無条件に行われたブランチに遭遇するが、BHTがこのブランチを示さない場合、BHTが誤ったことが知られる。この時点での適正なアクションは公式にそのブランチを実行することであり、しかも、新しいBHTエントリを作成し、ブランチの存在を示すことである。第2に、BHTが所定のアドレスで行われたブランチを示すが、このアドレスでデコードされた命令がブランチ命令でない場合、BHTがエラーであることが知られる。この時点での適正なアクションは、違反(offending)エントリをBHTから削除し、そのエントリの存在により影響された命令を事前に取り出す際にリダイレクションをアポートすることである。後者の例では、コードにブランチ命令がないときの命令事前取り出しをリダイレクトすることによるペナルティのサイクルを、BHTにより生じさせることができることに注意すべきである。

【0022】アドレス生成タイム(AGEN)にて、生成された目標アドレスがBHTにより予測され(しかも、プロセッサで待ち行列化され)た目標アドレスと同一でないか否かを、BWGは明らかにする。この時点での適正なアクションは、BHTエントリの目標アドレスを訂正し、誤った目標バスに向けられた命令事前取り出しを打ち切り、適正な目標バスに向けられた命令事前取り出しをリダイレクトすることである。

【0023】実行タイム(EXEC)にて、BWGの原因となることができるブランチのみが、条件付きブランチである。というのは、ブランチ条件のレゾリューション(resolution)がEXECの間でパフォーマンスされるからである。BHTがこのようなことを示さないとき、ブランチが行なわれることをEXECが判定する場合か、あるいは、ブランチが行われたことをBHTが示すとき、ブランチが行われないことをEXECが判定する場合に、BWGが生じる。いずれの場合でも、適正なアクションはBHTを更新し、ブランチの新しいアクションを示すとともに、新しいアクションに従って命令事前取り出しをリダイレクトすることである。

【0024】これまでのところは、命令を1つつつデコードするパイプラインプロセッサに説明を限定した。ブランチ命令はパイプラインフローを崩壊させる崩壊力であることは明白であり、しかも、ヒストリ型予測機構がブランチに起因する遅延の多くを削除することができることは明白である。

【0025】スーパースカラプロセッサでは、命令の多重性が同時にデコードされる(例えば、Johnson, M., Superscalar Microprocessor Design, Prentice Hall, Chap

ter1-2, (1991)を参照。この書名を記載して本明細書の一部とする)。この分野では、ブランチのパフォーマンス・インプリケーションはさらに厳しい。特に、サイクルごとに多重命令デコードを維持するため、サイクルごとの多重命令取り出しを維持することは必要である。ブランチは命令取り出しプロセスを不安定にする主要な原因であるので、ブランチを確実に予測することは、妥当なスーパースカラ設計の要件である。

【0026】さらにまた重要なことであるが、ほとんどのプログラムは平均して3つないし5つの命令に対して1つのブランチ命令を有する。これは平均であり、しかも、分散が必ずしも均一ではないので、1つないし2つの命令あたり1つのブランチを含むコードが頻繁に生じる。ブランチは汎用のプログラムで普及しているので、全く堅固なスーパースカラプロセッサは、2つ以上のブランチ命令を同時に予測することができなくてはならないし、適正な命令事前取り出しを開始することができなくてはならない。

【0027】命令取り出しに関係するようなブランチ予測の例は、the abstract of "Limits on Multiple Instruction Issue" by Michael D. Smith, Mike Johnson, and Mark A. Horowitz in computer Architecture News, Vol. 17, No. 2, April 1989, pp. 290-302 に示されている。この要約には、次のように記載されている。

【0028】高度に適正化され、非科学的なアプリケーションにて、サイクル当たり2つ以上の命令を実行する実行レートを維持することができるプロセッサの設計を限定するものを調査したものである。我々は、トレースドライブ型(trace-driven)シミュレーションを用いて、これらのアプリケーションがサイクル当たり約2つの命令の命令レートを維持するのに十分な命令独立を含む可否かを判定する。簡単なインプリメンテーションでは、コストに関して考察し、1サイクルで3つ以上の命令をデコードすることに全く反する結論を下している。このような制約がある場合、実行ハードウェアが複雑であるためというより、命令取り出し効率により、命令レベルで到達できる並列性が制限されるからである。

【0029】"Future Direction"セクションでは、著者は次のように結論している。すなわち、困難なことは、実際には、ブランチの周波数と、記憶装置における命令のランダムな位置合わせとを与える実行装置により必要とされる命令帯域を提供することにある。典型的なパイプラインRISC型プロセッサの取り出し帯域要件を維持するのに必要な技法は、充分な解法をスーパースカラアーキテクチャを提供しないことである。

【0030】命令取り出し帯域が多重デコードを制限するという結論は、"Single Instruction Parallelism is Greater Than Two," by Michael Butler, Tse-Yeh, Yale Patt, Mitch Alsup, Hunter Scales and Michael Shebanow, Computer Architecture News, Vol. 19, No. 3,

pp. 276, May 1991 でさらに実証されている。その要約を記述する。

【0031】最近の研究によると、少ない並列性(サイクル当たり2オペレーション未満)が単一命令ストリームで利用可能である。利用可能な並列性の量がプロセッサの設計に影響を与えるので、並列性がどのくらい実際に存在するかを検証するのは困難である。この研究では、資源の制約を異ならせて、SPECベンチマークの実行をモデル化している。我々は前の研究者の仕事を繰り返し、我々は、我々が課したハードウェア資源制約にもとで、同様の結果を得たことを示す。一方、プログラムの意味論により必要とされる制約を除いて、全ての制約が除去されるとき、並列性の度合いがサイクル当たり17個を超える命令であることを見出した。最後に、これは単一の命令ストリーム並列性を明らかにするには多分最も重要なことであるが、ハードウェアが適正に平衡を保っている場合、今日の妥当な設計によるプロセッサ上でサイクル当たり2.0 ないし5.8 命令を維持することができることを示す。

【0032】この研究では、著者はサイクル当たり1つのブランチのみを処理するが、"Conclusions" セクションでは、著者は、「サイクル当たり複数のブランチを行うことは、利用可能な並列性の量を増加させることである。」と述べている。さらに、次のように述べている。すなわち、サイクル当たり2.0 ないし5.8 命令という結論は、ウィンドウサイズと発行レートが制限されているが、今日妥当性があるものと見られた、制約されたデータフローエンジンから得られる。統合レベルと帯域幅が増加すると、それに対応して、ウィンドウサイズと発行レートが増加する。限界では、我々の境界のないウィンドウサイズと発行レートマシン(UDF) は、17 ないし1165 個の範囲でサイクル当たりの命令を示す。これは(まだ)可能ではないと我々は言っているが、我々は、サイクル当たり5命令を優に超す数値を期待している。

【0033】1ブランチを超えるブランチを予測し、しかも、サイクル当たり1ブランチグループを超えるブランチグループを取り出す必要があるのは明らかである。ブランチグループは1つの命令例であり、その命令アドレスは順次であり、すなわち、非非結合系列である。

【0034】そのため、全く堅固なスカルプロセッサは、同時に、1ブランチ命令を超えるブランチ命令を予測することができなくてはならないし、適正な命令事前取り出しを開始できなくてはならない。

【0035】

【課題を解決するための手段】条件付きブランチ命令は、プログラムの直線的実行を、前の計算の結果に基づき崩壊させる命令である。ブランチ命令が行われた場合、実行される次の命令は、実質的に後に来るブランチ命令ではない。その命令は、プログラムの別の部分から来ることになる。

10

20

30

40

50

【0036】サイクル当たり1命令のみをデコードする従来のプロセッサでは、ブランチ命令は次の2つの理由によりパフォーマンスに対して充分関わりがある。第1の理由は、プロセッサは、ブランチを解決するのに用いられる計算の結果を留保するのを遅延させるのが典型的である。第2の理由は、ブランチが行われたとき、命令取り出しと命令実行の間のパイプラインで崩壊が生じる。というのは、命令取り出しは名目上次の順次プロセスであるからである。命令ストリームで単一のブランチを予測することに関係する現技法を拡張したものが存在する。これは、命令の幾つかを、単一デコードプロセッサで減少させるか、あるいは削除させることができる。

【0037】「スーパスカラプロセッサ」の分野では、命令の多重性を同時にデコードする試みが行われる。ブランチは典型的なプログラムでは広範囲に広まっており、しかも、幾つかのプログラムでは2命令ごとにブランチを生じさせることができるので、スーパスカラプロセッサを構築するという試みは今日まで制限されてきた。

【0038】文献の調査により、次のことが分かった。すなわち、ごく特別な場合を除き、1サイクルごとに2または3命令を超える命令をデコードするスーパスカラプロセッサを実際にインプリメントすることが、ブランチ命令に固有のものによりできなくなるということは至極当然であることが分かった。この制限やその理由は至極当然であるが、これまでの問題に満足に答えることができる提案は何もなされてこなかった。多くの実践者はブランチ間の距離が多重デコーディングを根本的に制限するものとみなしている。

【0039】本発明に係る新規の機構は上記の問題点を解決するものである。本発明は、実行プログラムでサイクル当たり複数のブランチの結果を同時に予測する手段を提供する。予測されるブランチの多重性は、そのプログラムで生じるブランチの論理系列を形成する。本発明により、ブランチの全系列を並列に予測して、複数命令のデコーディングを単一のブランチグループに制限しななければならないという古い制限が取り除かれる。

【0040】その結果、本発明はスーパスカラ処理のパフォーマンスを改善すると共に、非常に高度の多重デコーディングに対する必須の発明である。すなわち、本発明により、以前は不可能と言われた新しいレベルのスーパスカラ処理が可能である。本発明は、論理的に系列を形成する複数ブランチを同時に予測する。本発明は、さらに、予測ブランチ系列を構成する複数非結合命令取り出しを開始する。

【0041】本発明は、信頼できる多重予測および多重事前取り出しを行うため、提供されなければならない2つの概念的な要素をインプリメントする。第1に、多重ブランチヒストリテーブル (MBHT) 機構は、複数ブランチグループの情報を提供するエントリを維持する。

すなわち、MBHTエントリは、各エントリの2つ以上のブランチ命令のブランチと目標アドレスをストアすることにより、ブランチグループの系列を記述する。第2に、ヒストリのブランチ予測は完全ではないので、予測されたブランチ系列の予測確率は系列の長さとともに幾何学的に減少する。そのため、本発明は、予測された系列への異なるオフセットと並列に探索され、予測された系列内のダイバーゼンスをできるだけ早く検知する複数MBHT機構によりその精度を高く維持する。

10 【0042】次のようにすることができる。

【0043】1) 本発明に係る方法は、複数のブランチ命令に対してそれぞれ未来目標アドレスを同時に予測する方法であり、前記複数のブランチ命令が前の実行で前のブランチ命令に順次従った方法であって、前記前のブランチ命令のブランチアドレスからストアアドレスを生成するステップと、マルチブランチヒストリテーブルのエントリであって、前記前のブランチ命令に対応するエントリであり、しかも、ブランチアドレスを含むとともに、前記各ブランチ命令の前の目標アドレスを含むエントリを前記ストアアドレスを用いてストアするステップと、前記エントリを前記マルチブランチヒストリテーブルから検索するステップと、前記各ブランチ命令に対して、前記エントリの前の目標アドレスを前記予測された未来目標アドレスとして適用するステップとを備えたことを特徴とする。

20 【0044】2) 本発明に係るシステムは、スーパスカラプロセッサの命令キャッシュに結合されたシステムであり、複数のブランチ命令を同時に予測し、前記予測を用いて、前記命令キャッシュに置かれる命令のブランチグループを取り出すシステムであって、(a) 第1の複数のブランチ命令の目標アドレスを、前記第1の複数のブランチ命令の実行ヒストリに基づき同時に予測する1次マルチブランチヒストリテーブル手段と、(b) 該1次マルチブランチヒストリテーブル手段に対して少なくとも1ブランチ命令だけ遅れ、第2の複数のブランチ命令の目標アドレスを、前記第2の複数のブランチ命令の実行ヒストリに基づき同時に予測し、前記1次マルチブランチヒストリテーブル手段による予測に際してダイバーゼンスを検知するために用いられる2次マルチブランチヒストリテーブル手段と、(c) 該2次マルチブランチヒストリテーブル手段と前記1次マルチブランチヒストリテーブル手段とに結合され、該1次マルチブランチヒストリテーブル手段による予測と、前記2次マルチブランチヒストリテーブル手段による予測とを比較し、ダイバーゼンスを検知する比較手段と、(d) 前記1次マルチブランチヒストリテーブル手段と前記2次マルチブランチヒストリテーブル手段とに結合され、前記1次マルチブランチヒストリテーブル手段と前記2次マルチブランチヒストリテーブル手段の前記実行ヒストリを、スーパスカラプロセッサにより実行されたブランチ命令の系列の

結果に基づき更新する更新手段とを備えたことを特徴とする。

【0045】3) 上記2)に記載のシステムにおいて、前記1次および第2マルチブランチヒストリテーブル手段は、それぞれ、マルチブランチヒストリテーブルを備え、該マルチブランチヒストリテーブルのエントリは、前に実行されたブランチ命令に対応し、前記前に実行されたブランチ命令の次の複数のブランチ命令のブランチおよび目標アドレスを前の実行から指定することを特徴とする。

【0046】4) 上記2)に記載のシステムにおいて、前記比較手段は、前記1次マルチブランチヒストリテーブル手段により予測された前記目標アドレスと、前記2次マルチブランチヒストリテーブルの対応する予測された目標アドレスとを比較する比較器を備えたことを特徴とするシステム。

【0047】5) 上記2)に記載のシステムにおいて、前記更新手段は、(e) 前記1次マルチブランチ予測手段による予測を表すアドレスを保持する1次保留予想待ち行列と、(f) 前記2次マルチブランチ予測手段による予測を表すアドレスを保持する2次保留予想待ち行列と、(g) 前記スーパスカラにより実際行われるブランチを表すアドレスを保持する更新待ち行列と、(h) 該更新待ち行列に保持された前記アドレスと、前記1次および2次保留予想待ち行列に保持された予測を表す前記アドレスとを比較し、前記1次および2次ブランチヒストリテーブルにそれぞれ対する不必要な更新をフィルタアウトする第2比較手段とを備えたことを特徴とするシステム。

【0048】6) 本発明に係るシステムは、複数のブランチ命令を同時に予測し、多重性の高い命令をデコードし実行するプロセッサより用いるためその予測を用いて複数の非結合命令系列を同時に取り出すシステムであって、(a) 前記プロセッサに結合され、少なくとも2つのブランチを行う命令の系列を記述するレコードを受信する更新待ち行列であって、前記レコードは少なくとも3つのアドレスを有し、前記複数のアドレスは前記複数のブランチを行う命令により接続された複数の非結合命令系列を識別し、前記複数のアドレスは第1部分および第2部分を含み、前記第1部分は開始アドレスであり、前記第2アドレスは残りの複数のアドレスであり、前記第1部分は第1命令系列を識別し、前記第2部分は前記複数の非結合系列の残りの複数の命令系列を識別する更新待ち行列と、(b) 前記更新待ち行列に結合されたマルチブランチヒストリテーブルであって、前記更新待ち行列からの前記レコードを受信し、しかも、前記レコードの前記第2部分を、将来使用するため、前記マルチブランチヒストリテーブルの場所であって、前記レコードの前記第1の部分により決定された場所にストアするマルチブランチヒストリテーブルと、(c) 前記プロセッサと

前記マルチブランチヒストリテーブルに結合され、前記プロセッサからの命令アドレスを受信する事前取り出しアドレスレジスタであり、前記レコードの前記前にストアされた第2部分の前記マルチブランチヒストリテーブルからの読み出しを、前記命令アドレスに対応してインエーブルにする事前取り出しアドレスレジスタであって、前記前にストアされた前記第2部分が前記プロセッサにより受信される事前取り出しアドレスレジスタと、(d) 前記マルチブランチヒストリテーブルと前記事前取り出しアドレスレジスタに結合された保留予想待ち行列であって、前記事前取り出しアドレスレジスタからの前記命令アドレスを、前記マルチブランチヒストリテーブルからの前記読み出しを用いて受信しストアする保留予想待ち行列であり、前記命令アドレスは前記読み出しを有するとともに、前記プロセッサにより前に実行された複数の非結合命令系列を識別する複数のアドレスよりなり、前記複数のヒストリのレコードである保留予想待ち行列と、(e) 前記保留予想待ち行列と前記更新待ち行列とに結合され、前記保留予想待ち行列からの前記ヒストリのレコードと、前記更新待ち行列により受信された前記レコードとを比較する比較器であって、前記保留予想待ち行列からの前記ヒストリのレコードと、前記更新待ち行列により受信された前記レコードとが一致しない場合にのみ、前記更新待ち行列により受信された前記レコードを前記更新待ち行列にストアすることができるシグナルを供給する比較器とを備えたことを特徴とするシステム。

【0049】7) 上記6)に記載のシステムにおいて、前記プロセッサからの前記命令アドレスと前記マルチブランチヒストリテーブルからの前記読み出しの部分を受信するセレクト手段であって、前記プロセッサからの前記命令アドレスか、あるいは、前記読み出しの前記部分かのをいずれかを選択し、しかも、前記選択を前記事前取り出しアドレスレジスタに経路指定するセレクト手段をさらに備えたことを特徴とするシステム。

【0050】8) 上記7)に記載のシステムにおいて、前記読み出しの部分は前記ブランチを行う命令のうちちょうど1つの目標アドレスであることを特徴とするシステム。

【0051】9) 本発明に係るシステムは、複数のブランチ命令を同時に予測し、多重性の高い命令をデコードし実行するプロセッサより用いるためその予測を用いて複数の非結合命令系列を同時に取り出すシステムであって、第1の複数のブランチを予測するための1次マルチブランチヒストリテーブルと、第2の複数のブランチを予測するための2次マルチブランチヒストリテーブルであって、前記第2の複数のブランチが前記第1の複数のブランチからの少なくとも1つのブランチ命令によりオフセットされる2次マルチブランチヒストリテーブルと、前記第1の複数の予測されたブランチが正しいか否

かを判定する手段と、前記第1の複数の予測されたブランチが正しい場合、前記第1の複数の予測されたブランチを選択する手段とを備えたことを特徴とするシステム。

【0052】10) 上記9)に記載のシステムにおいて、前記1次マルチブランチヒストリテーブルは、第1の1次ブランチグループと第2の1次ブランチグループをそれぞれ含むエン트리と、前記1次ブランチグループを、前記第1の1次ブランチグループからの目標アドレスか、あるいは、前記1次マルチブランチヒストリテーブルからの前記第2の1次ブランチグループからの目標アドレスに基づき、選択する手段とを備えたことを特徴とするシステム。

【0053】11) 上記10)に記載のシステムにおいて、前記2次マルチブランチヒストリテーブルは、第1の2次ブランチグループと第2の2次ブランチグループとをそれぞれ含むエン트리と、該エントリのうちの1つを前記1次マルチブランチヒストリテーブルから受信された目標アドレスに基づき選択する手段とを備えたことを特徴とするシステム。

【0054】12) 上記9)に記載のシステムにおいて、前記判定手段は、前記1次マルチブランチヒストリテーブルからの前記選択されたエントリの第2の1次ブランチグループと、前記2次マルチブランチヒストリテーブルからの前記選択されたエントリの第1の2次ブランチグループとを比較する手段とを備えたことを特徴とするシステム。

【0055】13) 上記9)に記載のシステムにおいて、前記選択手段は、前記1次マルチブランチヒストリテーブルからの前記選択されたエントリの前記第2の1次ブランチグループと、前記第2のマルチブランチヒストリテーブルからの前記選択されたエントリの前記第1の2次ブランチグループとを比較した結果、等しい場合、前記選択されたエントリを、前記1次マルチブランチヒストリテーブルから選択する手段と、前記1次マルチブランチヒストリテーブルからの前記選択されたエントリの前記第2の1次ブランチグループと、前記第2のマルチブランチヒストリテーブルからの前記選択されたエントリの前記第1の2次ブランチグループとを比較した結果、等しくない場合、前記選択されたエントリを、前記2次マルチブランチヒストリテーブルから選択する手段とを備えたことを特徴とするシステム。

【0056】14) 本発明に係る方法は、複数のブランチ命令を同時に予測し、多重性の高い命令を同時にデコードし実行するプロセッサにより用いるため、前記複数の予測を用いて、複数の非結合命令系列を取り出す方法であって、(1) 前記プロセッサに結合された更新待ち行列の少なくとも2つのブランチを行う命令を記述する記録を受信するステップであって、前記記録は少なくとも3つのアドレスを備え、前記複数のアドレスは前

記複数のブランチを行う命令により接続される複数の非結合命令系列を識別し、前記複数のアドレスは第1部分と第2部分を含み、前記第1部分は開始アドレスであり、前記第2部分は残りの複数のアドレスであり、前記第1部分は第1命令待ち行列を識別し、前記第2部分は前記複数の非結合命令待ち行列のうちの複数の残りの命令待ち行列を識別するステップと、(2) 前記更新待ち行列に結合され、マルチブランチヒストリテーブルの前記更新待ち行列からの前記記録を受信し、将来使用するため、前記記録の前記部分を、前記マルチブランチヒストリテーブルの場所であって、前記記録の前記第1部分により決定された場所にストアするステップと、(3) 前記プロセッサからの命令アドレスを事前アドレスレジスタで受信するステップであって、前記記録の前記前にストアされた第2部分の前記マルチブランチヒストリテーブルからの読み出しをイネーブルにし、前記前にストアされた第2部分が前記プロセッサにより受信されるステップと、(4) 前記事前取り出しアドレスレジスタからの前記命令アドレスを、前記マルチブランチヒストリテーブルからの前記読み出しとともに保留予想待ち行列にストアするステップであって、前記命令アドレスは、前記読み出しを有するとともに、前記プロセッサにより前に実行された複数の非結合命令系列を識別する複数のアドレスよりなるステップと、(5) 前記ヒストリの記録と前記更新待ち行列により受信された前記記録とを比較器で比較し、しかも、前期保留予想待ち行列からの前記ヒストリの記録と、前記更新待ち行列により受信された前記記録とが一致しない場合にのみ、前記待ち行列により受信された前記記録を前記更新待ち行列にストアすることができるシグナルを供給するステップとを備えたことを特徴とする方法。

【0057】15) 上記14)に記載の方法において、(6) 前記プロセッサからの前記命令アドレスと、前記マルチブランチヒストリテーブルからの前記読み出しの部分とを受信するステップと、(7) 前記プロセッサからの前記命令アドレスか、あるいは前記読み出しの前記部分のいずれかを選択するステップと、(8) 前記選択を前記事前取り出しアドレスレジスタに経路指定するステップとをさらに備えたことを特徴とする方法。

【0058】16) 上記15)に記載の方法において、前記読み出しの前記部分は、前記ブランチを行う命令のうちのちょうど1つの目標アドレスであることを特徴とする方法。

【0059】一般的に、 n 個のブランチグループを同時に予測しかつ事前取り出しを行うため、MBHTエンタリは n 個のブランチの系列についての情報を提供する。さらに、 n 個以下のMBHT機構は並列に探索される。 n 個のブランチよりなる幾つかの部分集合に対して、ブランチグループ当たり1個のMBHTである。並列探索は、複数のテーブルが予測されたストリーム内のダイバ

10

20

30

40

50

ーゼンスを示すとき、早期回復機構を提供する。

【0060】本発明は、3つの主な実施例を含む。(1) 第1の実施例は1重多重ブランチ予測装置に関するものであり、この装置により、1サイクルで複数のブランチグループを予測することができ、誤った予測を早期に検知することができる。(2) 第2の実施例は、n重多重ブランチ予測装置に関するものであり、この装置により、n個のブランチ命令の多重性を同時に予測することができ、誤った予測を早期に検知し、同様に、誤った予測を訂正することができる。(3) 第3の実施例は2重多重予測装置に関するものであり、この装置により、予測したストリーム内のダイバーゼンスを検知することができ、しかも、そのストリーム内でリダイレクション（訂正）を行うことができる。

【0061】第3の実施例、すなわち、2重多重ブランチ予測装置の構造およびオペレーションを要約して説明する。その2重多重ブランチ予測装置は、同時に、複数のブランチ命令を予測する。その装置はスーパーカラープロセッサの命令キャッシュに結合されている。その装置による予測を用いて、その命令キャッシュに置かれる命令のブランチグループが取り出される。

【0062】2重多重ブランチ予測装置は、次の要素を含む。すなわち、1次事前取り出しアドレスレジスタと、1次多重ブランチヒストリテーブルと、2次多重ブランチヒストリテーブルと、比較器と、更新待ち行列とを含む。

【0063】1次多重ブランチヒストリテーブルは第1の複数のブランチ命令の目標アドレスを、第1の複数のブランチ命令の実行ヒストリに基づき、同時に予測する。

【0064】2次多重ブランチヒストリテーブルは、1次多重ブランチヒストリテーブルに対して少なくとも1ブランチ命令だけ遅れる。2次多重ブランチヒストリテーブルは、第2の複数のブランチ命令の目標アドレスを、第2のブランチ命令の実行ヒストリに基づき、同時に予測する。さらに、2次多重ブランチヒストリテーブルを用いて、1次多重ブランチヒストリテーブルにより行われる予測のダイバーゼンスを検査する。

【0065】比較器は1次多重ブランチヒストリテーブルに結合され、しかも、2次多重ブランチヒストリテーブルに結合されている。比較器は1次多重ブランチヒストリテーブルによる予測と、2次多重ブランチヒストリテーブルによる予測とを比較し、ダイバーゼンスを検知する。

【0066】最後に、更新待ち行列は1次多重ブランチヒストリテーブルに結合され、しかも、2次多重ブランチヒストリテーブルに結合されている。更新待ち行列を用いて、1次多重ブランチヒストリテーブルと2次多重ブランチヒストリテーブルの実行ヒストリを、スーパーカラープロセッサにより実行されるブランチ命令の系列の結果に基づき更新する。

【0067】要約すると、本発明の主な特徴は、未来ブランチの多重性を同時に予測し、複数未結合プログラムセグメントの同時取り出しを可能にすることである。ブランチの多重性を予測することは、命令レベルの並列性に対する現制限を克服するために不可欠である。このことにより、コンピュータの最大処理速度が著しく速くなる。例えば、上記“Single Instruction Parallelism is Greater Than Two”に記載があるように、サイクル当たり2ないし5.8命令を実行するという現制限が除去され、考案したサイクル当たり17ないし165命令まで拡張される。

【0068】さらに、本発明は、正確でないブランチ予測のダイバーゼンスを検知し、ペナルティを課すことなくこのようなダイバーゼンスを訂正する。

【0069】本発明の上記目的および他の目的、特徴、および効果は、本発明の実施例の記述によりさらに明らかになるであろう。

【0070】

【実施例】

用語の定義

次に用語およびその用語に対応する意味を示す。

【0071】ブランチアドレス ブランチ命令が駐在するアドレス

目標アドレス ブランチ時に制御が渡されるアドレス

ブランチグループ アドレスが順次アドレスである命令の系列。すなわち、非結合系列である。各ブランチグループはブランチ目標命令から開始され、第1の行われたブランチに対して順次であり、かつ、第1の行われたブランチを含む全ての命令を継続する。

【0072】非結合 プログラム記憶領域内の順次記憶場所に必ずしも全てが駐在しない命令のグループをいう。

【0073】トリプル 3つの行われたブランチ、3つのブランチグループの系列、または、順番に実行される3つのブランチグループの開始アドレスの系列を用いるのに用いられる一般的な用語。

【0074】BWG(branch-wrong-guess) ブランチ予測機構がブランチ命令の結果を誤って予測するときの事象。

【0075】定常状態オペレーション 全ての予測が実行時に実際行われたブランチに対応する場合のオペレーション。

【0076】ENDOP 任意の命令実行の完了を特に言う。

【0077】I. 概観

本発明に係る実施例が3つある。第1の実施例は1重多重ブランチ予測装置に関するものであり、第2の実施例はダイバーゼンス検知およびリダイレクションを行うn重多重ブランチヒストリ装置に関し、第3の実施例は

ダイバージェンス検知およびリダイレクションを行う2重多重ブランチ予測装置に関する。

【0078】「本発明を実行するための最良のモード」の次にはappendixセクションがある。appendix Aは、2重多重ブランチ予測装置のサイクルごとのオペレーションを示すテーブルである。

【0079】ブランチグループはそのアドレスが順次アドレスである命令の系列である。すなわち、非非結合系列である。ブランチグループの第1命令はブランチ目標命令である。ブランチ命令により、命令順序付けはあるブランチグループから次のブランチグループに飛越す。

【0080】各ブランチグループはブランチ目標命令から開始し、第1行われたブランチを含むことができる目標命令に対して順次である全ての命令を継続する。そのブランチグループを終結するブランチは、新しいブランチグループに飛越すか、あるいは、同一のブランチグループ開始アドレスに飛越す。一般的な複数命令系列は一連のブランチグループを備えている。すなわち、一般的な複数命令系列は一連の非非結合命令系列を備えている。

【0081】図1は本発明がオペレートするコンピュータをベースとする環境を示す高レベルブロック図である。図1を説明する。スーパーカプロセッサ（「プロセッサ」）203は、各処理サイクルで、複数命令をデコードし実行することができる。従来例では、各サイクルでデコードすることができる命令の数は、ブランチグループ内の命令の数の制限された。本発明は、以下に説明するが、この制限を取り除いている。

【0082】長い複数命令系列は複数ブランチグループを一般的に含む。各サイクルでの長い複数命令系列のデコーディングを維持するため、各系列の構成要素ブランチグループを取り出すことが必要である。そのため、命令パスはブランチグループ全体を取り出させるほど充分に広くなければならない。さらに、構成要素ブランチグループを記憶装置の非結合場所から同時に取り出さなければならないので、マルチポートキャッシュ202は複数非結合命令取り出しを並列にサービスするのに必要である。マルチポートキャッシュ202の構造およびオペレーションは当業者に周知である。

【0083】本発明の第1の好ましい実施例は、マルチブランチ予測装置201の例である。マルチブランチ予測装置201はブランチグループに結合された複数ブランチをサイクルごとに予測する。また、マルチブランチ予測装置201はマルチポートキャッシュ202からのこれらブランチグループの取り出しを開始する。

【0084】プロセッサ203がプログラムを開始するか、あるいは、BWG(branch-wrong-guess)から回復すると、プロセッサ203はリスタートアドレスをアドレスバス214に供給する。これは、処理される命令ストリームの第1ブランチグループの開始アドレスである。そ

の開始アドレスはマルチポートキャッシュ202に伝送されるとともに、アドレスバス214を介してマルチブランチ予測装置201に伝送される。

【0085】第1ブランチグループは n (n は整数である)本の命令パスのうちの1本の命令パスを介してプロセッサ203によりマルチポートキャッシュ202から取り出される。これら n 本のパスはそれぞれ単一のサイクルでブランチグループ全体を伝送することができる。すなわち、バス212は極端に広域のな結合を提供する。

【0086】第1ブランチグループがマルチポートキャッシュ202から取り出されている間、バス214上のリスタートアドレスを用いて、マルチブランチ予測装置201の参照(look-up)に影響を与える。マルチブランチ予測装置201は第1ブランチグループの後に来る n 個のブランチグループを予測する。マルチブランチ予測装置201は単一のサイクルで n 個のグループを全て予測する。そして、各グループの開始命令アドレスを n 本のアドレスバス211を介してマルチポートキャッシュ202に供給する。

【0087】マルチブランチ予測装置201と、マルチポートキャッシュ202と、プロセッサ203は、パイプラインを構成することに注意すべきである。上述した第1サイクルでは、第1ブランチグループがマルチポートキャッシュ202から取り出され、一方、次の n 個のブランチグループがマルチブランチ予測装置201により予測される。第1ブランチグループはバス212のうちの1つのパスを介してプロセッサ203に伝送される。次の n 個のブランチグループのアドレスは、 n 本のアドレスバス211を介してマルチポートキャッシュ202に伝送される。

【0088】第2サイクルでは、第1ブランチグループの命令がプロセッサ203によりデコードされ、一方、次の n 個のブランチグループが全てマルチポートキャッシュ202から同時に取り出される。これら n 個のブランチグループはバス212を介してプロセッサ203に伝送される。同時に、マルチブランチ予測装置201は反復し次の n 個のブランチグループを予測する。

【0089】第3サイクルおよびその後の各サイクルでは、 n 個のブランチグループがプロセッサ203により同時にデコードされる。 n 個のブランチグループのサイクル当りのデコーディング率は、マルチブランチ予測装置201がブランチの予測を誤ったか否かをプロセッサ203が判定するまで、維持される。このとき、プロセスは、第1サイクルに関して上述したように、新しい命令をリスタートさせる。

【0090】プロセッサ203が命令の実行を完了すると、完了したブランチの情報をブランチ完了待ち行列204に置く。プロセッサ203は各サイクルでの n 個のブランチの系列の情報を待ち行列化することができる。

プロセッサ203はその情報をn本のバスを介してブランチ完了待ち行列204に伝送する。

【0091】ブランチ完了待ち行列(待ち行列)204は、情報をFIFO(first-in-first-out)で待機解除する。待ち行列204は2n個のエントリを保持することができなければならない。一連のn+1個のエントリは、単一のブランチの後にn個の連続ブランチグループが来ることを表すことに注意すべきである。そのため、n+k個のエントリが待ち行列化されたときは、常に、それらのエントリにより、n+1個のブランチグループのうちのk-1個の系列が記述される。

【0092】n+k個のエントリが待ち行列化されたときは、常に、n+1個のタブルのうちの関連するk-1個の系列が2n本のバス215のうちの部分集合を介してマルチブランチ予測装置201に転送される。転送が完了した後、待ち行列204は全体的に右にk-1桁だけシフトされる。その結果、k-1個の最も古いエントリは破棄される。待ち行列のオペレーションは当業者に周知である。

【0093】要約すると、図1はn個のブランチグループが各サイクルで予測される場合を示す。第1の好ましい実施例では、n=2に対して、すなわち、サイクルごとに2つのブランチを予測するマルチブランチ予測に対する特定の実施例に対して、マルチブランチ予測装置のオペレーションを詳細に説明した。n=2というポイントは本発明をデモンストレートする最も簡単な例として選択される。しかし、これが限度でないことは当業者にとって当然のことである。

【0094】II. プログラム例のシーケンス

図2は典型的なプログラムをシンボリックに示す。このプログラムはブランチグループを含む。ブランチグループはそれぞれ行われたブランチで終了する順次命令のshortグループである。各行われたブランチにより、新しいブランチグループの最初に飛越す。

【0095】例えば、シンボリックアドレスAから開始し、n個の命令を含むブランチグループが存在する。そのブランチグループの最後の命令は、シンボリックアドレスBから開始される新しいブランチグループにgoする行われたブランチである。

【0096】アドレスBからのb個の命令のオフセットには、アドレスCに行われる条件付きブランチが存在する。あるいは、条件付き命令は次の順序命令にフォールスルー(fall through)することができる。その条件付きブランチがフォールスルーする場合、アドレスBからn-1個の命令(アドレスxに移動される)分のオフセットに無条件ブランチが存在する。

【0097】このプログラムの他のブランチには次のものがある。すなわち、アドレスDから開始されるブランチグループから、アドレスEから開始されるブランチグループまで。アドレスXから開始されるブランチグルー

ブから、アドレスYから開始されるブランチグループまで。アドレスEから開始されるブランチグループから、アドレスAから開始されるブランチグループまで。アドレスYから開始されるブランチグループから、アドレスAから開始されるブランチグループまで。アドレスCから開始されるブランチグループから、アドレスDから開始されるブランチグループまで。

【0098】既に述べたように、本実施例はn=2である場合の例である。本装置のこの記述では、2個のブランチはサイクルごとに予測される。そして、この例の場合、値(n+1)-タブルは3-タブルであり、トリプル(triple)と呼ばれている。図2の右端カラムは、左カラムのプログラムが実行されたときに生じるブランチグループのトリプルに関係するアドレスをリストアップしたものである。

【0099】例えば、リストアップされた第1のトリプルはABCである。このトリプルは、アドレスAから開始されるブランチグループの後に、アドレスBおよびCからそれぞれ開始される一連の2つのブランチが来ることを表す。第2トリプルはBの後にCおよびDがそれぞれ来ることを表す。図示のトリプルの場合、アドレスB+bの条件付きブランチは、その最初の3つの実行で行われ、次の3つの実行では行われず、そして、再び行われるものと仮定する。そのため、プログラムフローは3回の反復でABCDEとなり、その後の3回の反復でABXYとなり、その後、再び、ABCDEとなる。このことは、トリプルABCはABXに変更され、トリプルBCDは3回の反復でBXYに変更され、ついで、両トリプルは元に戻される。残りのトリプル、すなわち、CDF, DEA, EAB, XYA, およびYABは変更されない。

【0100】マルチブランチ予測装置の次の記述は図2の例を用いてそのオペレーションを説明する。

【0101】III. 1重マルチブランチ予測装置

図3はサイクルごとに2つのブランチを予測することができる1重マルチブランチ予測装置のブロック図を示す。これは本発明を説明する最も簡単な装置である。この装置は、単一のテーブルを有するので「1重」と呼ばれている。この装置の単一のテーブルはマルチブランチヒストリテーブル(MBHT)401と呼ばれる。

【0102】MBHT401の各エントリは2つのブランチグループの系列についての情報を含む。MBHT401からの2つの出力411および412は対になっている。第1の出力411は第1ブランチグループを記述し、BG1で表す。第2の出力412は第2ブランチグループを記述し、BG2で表す。

【0103】各ブランチは2つのアドレスにより特徴付けられる。例えば、BG1は1対のアドレスBA1およびTA1を備えている。第1のアドレスはブランチ命令のアドレス、すなわち、BG1である。第2のアドレス

10

20

30

40

50

は対応する目標命令のアドレス、すなわち、TA1である。複合エントリ(BG1, BG2)は4つのアドレスを含み、対応するトリプルの第1のアドレスにより決定されるテーブルの場所にストアされる。図3のBG1およびBG2は、図1の命令アドレスIA_nに対応する。

【0104】例えば、図2のプログラムはトリプルABCを含む。このトリプルの実行により、エントリがMBHT401に作成されるであろう。そのエントリの場所はアドレスAにより決定されるであろう。この特定のプログラムの場合、トリプルABCに対応するエントリのフィールドは、

BA1 = A + n_a - 1; TA1 = B; BA2 = B + b; TA2 = C
である。

【0105】図2を説明する。MBHT401がアドレスAを用いて継続して探索される場合、このエントリが見付け出される。この探索を、組み合わせロジックを用いてPFAR402に保持されたアドレスと、MBHT401のエントリのアドレスフィールドと比較して、インプリメントすることができる。そのエントリは、アドレスA + n_a - 1に行われたブランチ(アドレスBにgoする)が存在し、アドレスB + bに別の行われたブランチ(アドレスCにgoする)が存在することを示す。従って、そのエントリは、AからBに遷移するとともに、BからCに遷移することを表すのに十分な情報を含む。すなわち、そのエントリは、トリプルABCを実行するのに十分な全命令の場所を正確に記述する。

【0106】図3に示すように、MBHT401は直接マッピングされるが、MBHT401をセットアソシアティブ構造体としてインプリメントすることができることは当然である。このようなインプリメンテーションは工業では至極当然である。さらに、このような構造体内には、必ずしも全てのBA1をストアする必要がないことは当然のことである。むしろ、BA1の一部を用いてその構造体内でエントリが突き止められる。よって、その一部はその場所に対して暗黙(implicit)である。さらに、BA2はTA1からのshort オフセットであることが分かっているので、BA2をオフセットとして表すことができることは、ルーティニア(routineer)にとって明らかである。これらは、テーブル空間のエコノミと関連しなければならない全てのインプリメンテーションとトレードオフである。これらトレードオフはどれも本発明にとって主要なものではない。MBHT401は事前取り出しアドレスレジスタ(PFAR)402のアドレスに基づき探索される。MBHT401が読み取られているとき、セクタ403はPFAR402のアドレスをMBHT401のアドレス入力端子413に経路指定する。PFAR402のアドレスは命令系列の開始アドレスである。そして、対応するエントリがMBHT401で見付け出された場合、出力BG1 411およびBG2 412は、命令系列で次の来るものと確信されるブラン

チグループのアドレスである。

【0107】エントリがMBHT401で見付け出された場合、信号線410, 411, および412上のアドレストリプルは、ブランチグループの予測されたトリプルを表す。予測されたトリプルは保留予想待ち行列404上に待ち行列化される。ただし、そのトリプルは不必要な更新をフィルタリングするために継続して用いられる。

【0108】図1から分かることであるが、トリプルはブランチ完了待ち行列204からマルチブランチ予測装置201にバス215を介して伝送される。これらのバス215は図3に示す信号線414, 415, および416としてマルチブランチ予測装置215に入る。すなわち、図3に入るこれらのバスはブランチグループトリプルのアドレスを伝送する。

【0109】完了されたトリプルがバス414, 415, および416上に到達すると、比較器はそれらのトリプルと、保留予想待ち行列404の予測されたトリプルとを比較する。保留予想待ち行列404が信号線414, 415, および416上にそれらのアドレスがある完了したトリプルを含まないことを、比較器405が判定した場合、トリプルはMBHT401に知られていないことを推測し、しかも、比較器405からの活動化信号417を受信した後、更新待ち行列406上に、到達トリプルを待ち行列化する。保留予想待ち行列404が信号線414, 415, および416上の完了したトリプルを比較器405が含まない場合、対応するエントリが保留予想待ち行列404から削除され、到達トリプルは更新待ち行列406上に待ち行列化されない。

【0110】更新待ち行列406に待ち行列化されたエントリがある場合は、それらのエントリは、MBHT401が探索されないサイクルで、MBHT401に伝送される。これらのエントリを転送するには、セクタ403はトリプルの第1アドレスをMBHT401のアドレス入力端子413に経路指定する。そして、トリプルの残りのアドレスはそれらのエントリがストアされたMBHT401のデータ入力端子419および420を介してMBHT401に経路指定される。

【0111】定常状態オペレーションでは、全予測が訂正され、図3の装置が動作する。すなわち、ブランチトリプルが正確に予測される限り、各トリプル412の第3アドレスはセクタ407(このセクタは2:1 マルチプレクサとして動作する)を介して経路指定され、次のサイクルでMBHT401を探索するため、その第3アドレスが開始アドレスとして用いられるPFAR402にロードされる。

【0112】図1から分かるように、プロセッサ203はプログラムを開始するか、あるいは、BWGから回復するとき、リスタートアドレスをアドレスバス214上に供給する。このバス214はマルチブランチ予測装置

201に接続され、図3に示すSTART 入力端子408に対応する。図3を説明する。START 入力端子408は処理される命令ストリームの第1ブランチグループの開始アドレスである。

【0113】そのため、パワーオンシーケンスの後か、あるいは、BWGの後に、セレクト407がセットされ、START 入力端子408をPFAR402に経路指定する。一度、その装置がスタートアップすると、定常状態オペレーションで既に述べたように、セレクト407が切り換えられ、出力412をPFAR402に経路指定する。(切り換えは種々の方法で行われる。例えば、プロセッサは図1に示すように制御信号をリスタートアドレスバス214を介してセレクト407に送信する。他の方法も可能であり、当業者には既に明らかであろう。)例えば、図2のプログラムが図3の装置と対話するときの図2のプログラムを考察する。図2のプログラムがアドレスA(受信アドレスAであるべきである)から開始されると、セレクト407により、ATART 入力端子408からPFAR402にロードされる。MBHT401が探索される。そして、トリプルABCがMBHT401にストアされ、出力411および412がそれぞれアドレスDおよびEを生成するものと仮定する。

【0114】第3サイクル(クロックサイクルまたはプロセッササイクル)では、PFAR402にアドレスEがロードされ、上述したようにオペレーションが継続し、連続したサイクルで、出力対AB, CD, EA, BC等を生成する。定常状態オペレーションは、BWGが存在するまで継続する。BWGの後、その装置はSTART 入力端子408からリスタートされる。そのオペレーションの詳細な説明は、MBHT401の更新を含めて、この後のセクションで行う。

【0115】IV. n 重マルチブランチ予測装置

本発明は、信頼できる多重予測を行うために提供される2つの要素を備えている。第1に、明らかなことであるが、多重予測装置はブランチ予測の多重性を同時に提供する。本発明のこの態様は既に前のセクションで述べた。第2に、多重予測機構が正確に信頼できる程度にパフォーマンスするには、その装置はブランチ予測の多重性のダイバーゼンスを検知しかつ訂正するための要素を含むべきである。

【0116】ブランチの態様を過去のヒストリに基づき予測する装置は、エラーがほとんどなく予測することができるが、完全にはできない。同時に1つのブランチを予測する装置では、予測誤りにより精度が容易に低下する。多重予測機構では、その精度は予測される系列の長さに対して幾何級数的に低下する。すなわち、予測誤りを行うと多重予測機構に多大な影響を与える。

【0117】図3を参照してこれまでに説明したことは、 $n=2$ すなわち1重の場合の装置に限定したが、その装置は第2の予測されたブランチのダイバーゼンスを予

測する能力はない。図3を次のように一般化することができる。すなわち、 $\langle BA1, TA1 \rangle, \langle BA2, TA2 \rangle, \dots, \langle BAN, TAN \rangle$ の n 個のアドレス対を含むために、MBHT401のエントリを拡張してサイクルごとに n 個の予測を行う装置に一般化することができる。このように一般化しても、依然として、予測された n -タブル内のダイバーゼンスを検知する能力に欠ける。

【0118】図4は汎用の n 重マルチブランチ予測装置のブロック図を示す。これはサイクルごとに n 個の予測を行い、しかも、 n 個の予測をそれぞれ検知し訂正する手段を提供する。図4は本発明の要素を遂行する最も汎用的なインプリメンテーションを示す。

【0119】図4は図3に示すマルチブランチ予測装置の n 個のコピーを含む。それらの装置はそれぞれ上述した n 個の同時予測を行うように拡張される。

【0120】特に、図4のブロックMBHT $n, \dots, MBHT2, MBHT1$ は、図3に示す装置を含む。 n 個のアドレス対を含むため、MMBHT401のエントリが拡張される。さらに、セレクト407を除去するとともに、その装置からのフィードバックバス412を除去し、しかも、1次入力408を直接PFAR402に経路指定して、図3の装置を図4で用いるように修正する。この入力408は適正な方法で図4の残りの回路によりドライブされる。図4に示す n 重マルチブランチ予測装置の動作を次に説明する。

【0121】図1から分かるように、プロセッサ203がプログラムを開始するか、あるいはBWGから回復されると、プロセッサ203はリスタートアドレスをアドレスバス214に供給する。マルチブランチ予測装置201に行くこのバス214は、図4のSTART 入力端子510である。これは、処理される命令ストリームの第1のブランチグループの開始アドレスである。

【0122】そのため、パワーオンシーケンスの後か、あるいは、BWGの後に、セレクト501はセットされ、START 入力510を、MBHT $n502$ に含まれる(図示しない)PFARに経路指定する。一度、その装置がスタートアップすると、セレクト501が切り換えられ、出力命令アドレス $IAN511$ をMBHT $n502$ に経路指定する。その結果、そのテーブルは図3で述べたように自己ドライブ型(self-driven)になる。

【0123】MBHT $n502$ は「1次マルチブランチヒストリテーブル」と呼ばれる。その役割は、 n 個のブランチであるという予測を前の予測よりさらに深く命令ストリームに入れることである。これは、第 n 番目の予測された命令アドレス $IAN511$ が各サイクルで1次テーブル502の入力に経路指定されて戻されるからである。

【0124】1次MBHT $n502$ がブランチの予測を決して誤らない場合、 $n-1$ 個の残りのMBHTとセレクトは必要ではない。すなわち、図3は図4の全ハードウェ

ア(ただし、入力セクタ501と1次MBHT502を除く)を除去することにより図4から直接得ることができる。 $n-1$ 個の残りのMBHTと図4の選択ロジックにより、早期に予測され、しかも、1次MBHT予測装置502から予測されるブランチ系列のダイバーゼンスが訂正される。

【0125】図4の出力は n -タブルの予測された命令アドレス512であり、IA1, IA2, ..., IAnのラベルが付されている。命令アドレスIA_n 511は n -タブルの n 番目のアドレスである。命令アドレスIA_n 511を用いて、上述したように、1次MBHT502をドライブする。 $n-1$ 個の残りの命令アドレスは異なる2次MBHTをドライブする。

【0126】例えば、命令アドレスIA1 513は2次MBHT1 503をドライブする。IA1 513の元のソースは1次MBHT502により予測された n -タブルの第1のブランチ目標アドレスであることに注意すべきである。2次テーブルMBHT1 503の出力は、IA1 513に連続すると予測されるブランチ目標アドレスの n -タブルである。

【0127】同時に、命令アドレスIA2 514は2次MBHT504をドライブする。IA2 514の元のソースは1次MBHT502により予測された n -タブルの2次ブランチ目標アドレスである。2次テーブルMBHT2 502の出力は、IA2 514に連続すると予測されたブランチ目標アドレスの n -タブルである。

【0128】比較器(CMP2)505は、アドレスIA2 514と、2次MBHT1 503の第1命令アドレス出力515と比較する。2次MBHT1 503はIA1 511によりドライブされるので、その第1命令アドレス出力515は1次MBHT502により予測されるように、IA2 514と同一にすべきであることを思い出して欲しい。

【0129】すなわち、MBHT1 503の第1出力515は、BMHT_n 502の第2出力と同一にすべきである。

【0130】同様に、MBHT1 504の第1出力516は、BMHT_n 502の第3出力と同一にすべきである。このことが比較器506により判定される。 $n-1$ 個の2次MBHTはそれぞれ比較器を有し、これらの比較器により、関連する2次MBHTにより予測された n -タブルの第1出力アドレスと、1次MBHT502により前のサイクルで予測された n -タブルの適正なアドレスとが比較される。全MBHTが一貫性のある情報を含む場合、比較器の全出力により、全アドレスが一致することが示される。

【0131】比較器のなかに一致しないものがある場合、その比較器と関連する2次MBHTは、1次MBHT502と矛盾する情報を含む。このとき、2次MBHT

Tの情報は正しいものと仮定され、1次MBHT502の情報は正しくないものと仮定される。

【0132】この仮定は、部分的には、このサブセクションの第3パラグラフで行った議論に基づくものである。すなわち、各テーブルの第1出力は、本来、第 k

(それぞれ連続している)出力より正確である。2次MBHTの場合、それぞれ、適正な出力と比較されるのは、第1出力であって、1次MBHT502の第1出力ではない。従って、誤って比較した場合には、2次MBHTには1次MBHTより高い優先順位が与えられる。

【0133】エンコード507は誤って比較をしたか否かを判定する。誤って比較をしない場合、エンコード出力518により、セクタ508は1次MBHT502の出力をレジスタ509に経路指定する。レジスタ509には次のサイクルで用いるため1次MBHT502の出力がストアされる。

【0134】誤った比較がちょうど1つだけある場合、エンコード507により、セクタ508は関連する2次MBHTの出力をレジスタ509に経路指定する。例えば、比較器505はMBHT503の第1出力が前のサイクルで予測したIA1 513と同一でないと判定した場合、MBHT1 503の出力をセクタ508を介してレジスタ509に経路指定する。

【0135】誤った比較が複数個ある場合、誤って比較するMBHTのうちの第1MBHTに優先順位が与えられる。というのは、そのMBHTは、予測されたストリームのダイバーゼンスの第1のポイントを示すからである。2次MBHTのうちの第1MBHTは、図4に示すように、右端のテーブルである。例えば、比較器505および506が両方とも誤って比較したことを示す場合、エンコード507はMBHT1 503に優先順位を与えることになる。

【0136】選択された n -タブルを保持するレジスタ509が n 個のMBHTのPFARに関して冗長であることに注意すべきである。すなわち、レジスタは n 個のPFAR内容を保持する。このレジスタは図中に存在する必要はないが、その後のサイクルで行われる予測を明確に区別する方法を示す。

【0137】また、図4に示す"UPDATES"入力519は完全をきするため含まれる。これはバスの集合であり、これらのバスを用いて、実行された $(n+1)$ -タブルをMBHTにストアするために伝送する。すなわち、これらは図1に示すバス215である。更新オペレーションはこのサブセクションでは説明していないが、前のサブセクションで説明した。また、次のサブセクションでさらに詳しく説明する。

【0138】2次MBHTは2つの機能を有する。第1には、 n -タブルの第1アドレス出力は、1次MBHTにより予測された n -タブルのダイバーゼンスを検知する手段を提供する。たとえば、MBHT1 503の出

力515はIA2 514と比較され、1次MBHT502により予測された第2アドレスがダイバースを判定する。

【0139】2次MBHTにより提供された第2機能は、第1アドレスが1次MBHTからの関連するアドレスと一致しない場合に、 n -タプルと置換することができる機能である。例えば、MBHT1 503は、 n -タプル515の第1アドレスが1次MBHT502からのIA1 514と一致しない場合、 n -タプル出力512を供給する。

【0140】すなわち、2次MBHTの第1アドレス出力がダイバースを検知し、残りの $n-1$ 個のアドレスがそのダイバースを直ちに訂正する。そのため、最後の $n-1$ 個のアドレスは、ダイバース検知のみを必要とするインプリメンテーションでは必要でない。一度、ダイバースを検知されると、その後のサイクルで、1次MBHT502によりダイバースの訂正を行うことができる。インプリメンテーションをより経済的であるが、僅かに遅くした場合、そのインプリメンテーションは単一のアドレスのみを出力する2次MBHTを有することになるであろう。

【0141】その設計の面では、2次MBHTの幾つかを削除することになる。すなわち、 n -タプルのダイバースをブランチごとに検知する必要はなくなる。2ブランチごとにダイバースを検知するだけで充分である。このことは、全くコストパフォーマンスとのトレードオフに過ぎない。汎用の k 重多重予測装置は $k < n$ から構成される。 $k = 1$ である場合、ダイバースの検知はない。

【0142】情報は全て同一のプロセッサから来るので、理論的には、全てのMBHTは全く同一の情報を含むべきであることに注意すべきである。このようになっている場合は、誤って比較することは全くなく、しかも、2次MBHTは必要でない。実際、MBHTは有限であり、セッター結合を制限したし、ポーティング等を制限した。異なったMBHTは全く同一の情報をストアすることができなくなる。さらに、実行されたストリームの中断に起因して、実行されたストリームのブランチを誤って予測するので、不完全な情報が更新パス519上に出力されることになる。すなわち、不完全なタプルをMBHTの幾つかにストアすることは、賢明なことである。特に、2次MBHTが全 n -タプルをストアしない場合は、上述したようにそうである。実際には、MBHTは全く同一の情報を含まない。情報の相違の程度はインプリメンテーションに依存する。

【0143】図5は多重予測の高レベルのロジックフローを示す。ステップ601では、新しい命令系列の第1アドレスを用いて、1次MBHTを探索する。この探索の結果、 n 個の予測アドレスが生成される。これら n 個のアドレスが次の3つのステップ602、603、およ

び604により同時に用いられる。

【0144】ステップ602にて、前のステップ601により予測された n -タプルの最後のアドレスは、1次MBHTの次の探索で用いられる、すなわち、第1ステップ601の次のサイクルで用いられる次の開始アドレスとしてステージングされる。同時に、ステップ603にて、残りの $n-1$ 個のアドレスを用いて、2次MBHTを探索する。また、同時に、ステップ604にて、前のステップ601にて予測された n 個のアドレスを用いて、 n 個の命令ストリームを取り出し、プロセッサによる実行のためにステージングする。

【0145】判定ステップ605にて、前のステップ603にて獲得された2次MBHTの出力は、前のステップ601にて1次MBHTから獲得された出力と比較される。予測されたストリームに矛盾がある場合は、次の開始アドレスがステップ606にて変更される。次の開始アドレスは、ステップ605にて判定した結果、相違するMBHTのうちの第1MBHTにより予測されたアドレスである。

【0146】一方、ステップ607にて、プロセッサは、前のステップ604にて取り出された命令ストリームをデコードし実行する。ブランチを実際に実行した結果、前のステップ601にて予測されたブランチの系列と相違する場合、次の開始アドレスが、処理ステップ607の実際に実行された系列により判定される。アドレスの置換が必要な場合は、選択ステップ608にて行われる。

【0147】最後のステップ608にて選択されたアドレスを用いて、プロセスを最初のステップ601からリスタートする。

【0148】V. 2重マルチブランチ予測装置
セクションIIIでは、多重予測を詳細に説明した。1重多重予測は最も簡単な多重予測であるが、ダイバース検知およびリダイレクションという態様を含まない。セクションIVでは、 n 重多重予測の概要を説明した。これは充分なダイバース検知と訂正を含む最も総称的な多重予測機構である。

【0149】このセクションでは、2重多重予測を詳細に説明する。2重多重予測は n 重予測の最も簡単な例である。すなわち、詳細に説明できる程度に簡単であるが、ダイバース検知を説明するには非常に複雑である。

【0150】図6はサイクルごとに2つのブランチを予測することができる2重多重予測装置のブロック図を示す。これは予測された対の第2のブランチのダイバースを検知しかつ訂正することができる。

【0151】図6は図3に示す1重多重予測機構の2つのコピーと、ダイバース検知のために用いられる比較器718と、予測されたストリームをリダイレクトするの用に用いられる2つのセクタ709および710と

を含む。2つの1重多重予測機構は、それぞれ、1次および2次マルチブランチヒストリテーブル、すなわち、PMBHT701およびSMBHT711と呼ばれる。

【0152】図3と図6とを比較して、異なるMBHTの要素を明らかにするのは有用である。第1に、図3の実際のMBHT401は2つのMBHT、すなわち、上述したが、図6には701および711として図示されている。図3のMBHT401を探索するために用いられるPFAR402は、図6には、1次および2次PFARとして図示され、それぞれ、PFAR702およびPFAR712とラベルが付されている。図3のセクタ403は、図6では、セクタ703および713である。図3の保留予想待ち行列(pending guess queue)404は、それぞれ、図6では、1次および2次保留予想待ち行列704および714として図示されている。図3の保留予想待ち行列に関連する比較器は、図6では、2つの比較器705および715として図示されている。最後に、図3の更新待ち行列406は、図6では、1次および2次更新待ち行列706および716として図示されている。図3の開始入力セクタ408は、図6の開始入力セクタ708と同一である。1次MBHTのみがこのセクタから入力を受信するn重予測の議論を思い出して欲しい。図6の新しいハードウェアは、既に述べた比較器718と、2つのセクタ709および710である。次に、図6に示す装置の動作を説明する。

【0153】図1から分かるように、プロセッサ203はプログラムを開始するか、あるいはBWGから回復するとき、リスタートアドレスをアドレスバス214上に供給する。マルチブランチ予測装置201に行くこのバス214は、図6に示すSTART入力719である。これは、処理される命令ストリームの第1ブランチグループの開始アドレスである。

【0154】そのため、パワーオンシーケンスの後か、あるいは、BWGの後に、セクタ708がセットされ、START入力719をPPFAR702に経路指定する。一度、その装置がスタートアップされると、セクタ708が切り換えられ、出力726をPPFAR702に経路指定し、定常状態動作を行う。

【0155】定常状態動作では、2重マルチブランチ予測装置、すなわち、BG1725の第1アドレス出力は、SPFAR712にロードされる。SPFAR712では、第1アドレス出力を用いて、2次MBHT711をその後のサイクルでドライブする。それと同時に、装置BG2726の第2アドレス出力がPPFAR702にロードされる。PPFAR702では、その第2アドレス出力を用いて、1次MBHT701をその後のサイクルでドライブする。

【0156】第2アドレス726を用いて、1次MBHT701をドライブすることにより、1次MBHT70

1は、最後に予測された対の前に2つのブランチをステータスさせることができる。すなわち、1次MBHT701はサイクルごとに2つの新しいブランチを予測する。第1アドレス725を用いて2次MBHT711をドライブすることにより、2次MBHT711は、1ブランチだけ1次MBHT701より遅れる。

【0157】そのため、2次MBHT711の第1出力723は、前のサイクルからの1次MBHT701の第2出力722と一致すべきである。しかし、前のサイクルからの1次MBHT701の第2出力722は、そのサイクルの終りでPPFAR702にストアされた。そのため、比較器718は2次MBHT711の第1出力723からのアドレスと、PPFAR702のアドレスとをサイクルごとに比較する。

【0158】これらのアドレスが一致するか否かを比較器718が判定する限り、1次MBHT701は引き続き2重マルチブランチ予測装置をドライブする。特に、比較器718により、セクタ709および710は、それぞれ、2重マルチブランチ予測装置700の出力725および726を経路指定する。上述したように、第1出力725を用いて2次MBHT711がドライブされることになり、第2出力726を用いて1次MBHT701をサイクルごとにドライブすることになる。

【0159】そのため、2次MBHT711の第1出力723は、前のサイクルからの1次MBHT701の第2出力722と一致すべきである。しかし、前のサイクルからの1次MBHT701の第2出力722は、PPFAR702にそのサイクルの終りでストアされることに注意すべきである。そのため、比較器718は、2次MBHT711の第1出力723からのアドレスと、PPFAR702のアドレスをサイクルごとに比較する。

【0160】これらのアドレスが一致するか否かが比較器718により判定される限り、1次MBHT701は2重マルチブランチ予測装置を引き続きドライブする。特に、比較器718により、セクタ709および710は、1次MBHT701からの出力721および722を、2重マルチブランチ予測装置700のそれぞれの出力725および726に経路指定する。上述したように、第1出力725を用いて2次MBHT711をドライブすることになる。しかも、第2出力726を用いて1次MBHT701を次のサイクルでドライブすることになる。

【0161】2つのアドレスが一致していないことが比較器718により判定された場合、2次MBHT711は2重マルチブランチ予測装置を次のサイクルでドライブすることになる。特に、比較器718により、セクタ709および710は2次MBHT711からの出力723および724を、2重マルチブランチ予測装置700のそれぞれの出力725および726に経路指定する。

【0162】これが、2重マルチブランチ予測装置700の動作の基本的な記述である。テーブルPMBHT701およびSMBHT711に共にパス(S1, S2, S3)727からのブランチグループトリプルの情報がロードされる。

【0163】テーブルPMBHT701およびSMBHT711の更新は、1重マルチブランチ予測装置400で説明したように行われる。非常に簡単に言うと、MBHT701, 711が予測すると、その予測を保留予想待ち行列704, 714に記録する。実行されたトリプルがパス727に到達すると、それらのトリプルは保留予想待ち行列706, 716の予測されたトリプルと比較される。実際のトリプルが予測されたトリプルと一致しない場合、実際のトリプル更新待ち行列706, 716に置かれる。更新待ち行列のエントリは遊休サイクルの間にMBHT701, 711に書き込まれる。

【0164】理論的には、MBHT701および711が共に無限に大きい無限連想テーブルである場合で、しかも、全トリプルを完全かつ適正に受信しストアする場合、2次MBHT711は必要ではない。実際には、MBHT701および711は必ずしも一貫性のある情報を含むわけではないので、予測されたストリームにダイバーゼンスが生じる。

【0165】VI. 2重マルチブランチ予測機構の動作例
このセクションでは、図2に示すプログラムがランしている間の図7に示す装置の動作を詳細に説明する。プログラムフローでは、図2において、アドレスB+bのブランチ命令が最初の6回の反復でとられるが、その後はとられないものと仮定する。そして、実行されるブランチグループは、ABCDEの6回の反復であり、その後のABXYの複数回の反復である。

【0166】図6は、構成要素待ち行列の正確な動作や、プロセッサパイプラインの正確な動作を伝える程詳細ではないことに注意すべきである。これらの正確な作業により、装置の動作の正確な方法を判定するので、作業を指定することが必要である。言い換えると、待ち行列は一般的に図6で参照され、しかも、特定のアプリケーションシステムのためにカスタマイズすることができる。正確な規則のうちの任意の集合を指定することができる、しかも、正確な集合はその装置の一般的な動作を変更することができる。しかし、基本的な発明は依然として変わらない。

【0167】説明のため、次の8つの規則が異なるカテゴリで用いられる。

【0168】プロセッサパイプラインの動作

・規則1—ブランチ命令が実行されたとき、その目標アドレスが、そのブランチの実際の実行サイクルで命令取り出しのためにステージングされる。

【0169】・規則2—ブランチグループは実行サイクル直後のサイクルで実行されたと考えられる。すな

わち、ブランチグループを実行して完了された任意のトリプルS1, S2, S3は、ブランチグループの実行直後のサイクルで2重多重予測装置に送信されることになる。

【0170】・規則3—1次MBHT705がミス(miss)されたサイクルで2次MBHT711がヒット(hit)される場合、2次MBHT711の出力がステージングされ、次のサイクルで命令をドライブする。

【0171】更新待ち行列の管理とMBHTの更新

10 ・規則4—重複エントリが更新待ち行列706, 716に置かれるのを阻止するものと仮定する。例えば、1次更新待ち行列706がトリプルABCを含み、しかも、トリプルABCの新しいコピーが更新待ち行列727に到達する場合、その新しいコピーは1次更新待ち行列707に置かれぬ。これは、エントリABCが1次保留予想待ち行列704にないとしても、その新しいコピーは1次更新待ち行列707に置かれぬ。

20 【0172】・規則5—プロセッサ203が2つのブランチグループを同時に実行すると、その結果、一般的には、2つのトリプルが2重多重予測装置700に同時に伝送される。更新待ち行列706, 716がサイクルごとに1つのトリプルのみを処理することができるものと仮定し、しかも、第2トリプルを保持するバッファはないものと仮定する。そして、2つのトリプルが同時に到達した場合は、第1のトリプルは1次更新待ち行列706にのみ送信され、第2のトリプルは2次更新待ち行列716にのみ送信される。

30 【0173】・規則6—名目上は、MBHT701, 711がサイクルごとに探索され、しかも、探索と同時に更新を行うことができない。そのため、更新待ち行列706, 716の更新が結局MBHT701, 711に置かれることを保証するため、ミスした後のサイクルで1つのサイクルに対して、更新に優先順位が与えられる。

【0174】保留予想待ち行列の管理

40 ・規則7—トリプルS1, S2, S3が更新パス727に到達し、かつ、保留予想待ち行列704, 714の全エントリと比較されると、少なくとも1つが保留予想待ち行列704, 714で一致した場合、最も古いエントリが削除され、そのエントリより古い全てのエントリはその待ち行列から削除される。

【0175】・規則8—ブランチが不正確に予想されたことがプロセッサ203により発見されたとき、保留予想待ち行列704, 714のエントリは全て削除される。

【0176】APPENDIX Aはサイクル(最初の56サイクルのオペレーション)ごとの図6に示す2重多重予測装置700のオペレーションを示す。2重多重予測装置700は、上記の規則を条件として、図2のプログラムをランしている。右端のカラムは0から始まり、55までのサイクルの番号を示す。残りのカラムには4つの主要

な区分がある。カラムの第1のセットはスーパーカラムプロセッサ203内のフローと、マルチポートキャッシュ内のフローを示す。アドレスステージのカラムは次のサイクルで取り出すためステージングされた命令アドレスを示す。キャッシュアクセスのカラムはそのサイクルの間に取り出されたブランチグループを示す。プロセッサオペレーションは2つのカラムに示す。すなわち、デコードおよび実行のカラムに示す。これらのカラムは対応するサイクルの間にデコードされ実行されるブランチグループを示す。ENDOPグループトリプルは2つのカラムを示す。というのは、上記規則5によれば、サイクルごとにENDOPするトリプルが2つ存在することができるからである。ENDOPトリプルは、上記規則3によれば、トリプルの最後のブランチグループの実行の後のサイクルで現れる。

【0177】カラムの残りの3つの区分は、2重マルチブランチ予測装置700内の資源を示す。それらの区分のうちの2つは1次MBHT701と2次MBHT711に対するものである。これら2つの区分の間の比較器のカラムは、2次MBHT711の第1出力と、PPFAR702の内容とを比較したときの比較器718（図6に示す）のオペレーションを示す。PPFAR702は名目上は前のサイクルからの1次MBHT701の第2の出力を含む。

【0178】各MBHT701, 711のオペレーションは1つのMBHTに対して7つのカラムで詳細に示す。それらのカラムのうちの第1のカラムはPFAR702, 712であり、MBHT701, 711を探索するのに用いるアドレスを示す。次の2つのカラム、すなわち、BG1 721, 723と、BG2 722, 724は、PFAR702, 712の内容を用いて、MBHT701, 711を探索した結果の2つの出力である。出力721, 722, および723, 724は、それぞれ、PFAR702, 712をロードした後のサイクルで利用可能である。残りの4つのカラムは4つの待ち行列704, 706, 714, 716のオペレーションを示す。

【0179】保留予想待ち行列704, 714と、更新待ち行列706, 716に対して、待ち行列に置かれるエントリを示す待ち行列化カラムと、待ち行列から離れるエントリを示す待機解除カラムがある。典型的には、対応するMBHT701, 711出力が現れるサイクルで、トリプルが保留予想待ち行列に待ち行列化される。トリプルがプロセッサ203でENDOPすると、保留予想待ち行列704, 714からの一致したエントリの待機が同一のサイクルで解除される。一致したエントリがない場合、ENDOPトリプルが上記規則5により更新待ち行列706, 716に待ち行列化される。すなわち、MBHT701, 711ごとに1つのトリプルが待ち行列化される。

【0180】MBHT701, 711が更新を受け取ると、更新が行われる同一のサイクルで、対応するエントリが保留予想待ち行列704, 714から待機解除される。ENDOPトリプルが受信されたとき、MBHT701, 711が遊休状態である場合、そのトリプルは、MBHT701, 711が更新されると、効率的に、同時に待ち行列化されかつ待機解除される。このことは、APPENDIX Aの待ち行列化カラムおよび待機解除カラムに共にトリプルをカッコで囲んで示す。例えば、APPENDIX Aの2次更新待ち行列カラムのサイクル#7を参照されたい。

【0181】この例のフローでは、プログラムの第1ブランチグループには開始アドレスAがある。そのフローは命令取り出しを述べたサイクル#0で命令アドレスAから開始される。また、ステージングを行って命令を取り出すアドレスを用いて、1次MBHT701を探索し、サイクル#0でPPFAR702にアドレスAがロードされる。この例では、MBHT701, 711は、共に、MBHT701, 711内の有効なエントリから開始されないものと仮定する。

【0182】サイクル#1にて、キャッシュがアクセスされ、ついで、アドレスAのブランチグループが取り出される。また、1次MBHT701の出力は利用可能になる。MBHT701は空であるので、探索してもミスする。PBG1 721およびPBG2 722カラムを参照されたい。ミスがない場合は、マルチブランチ予測装置700は命令取り出しのドライブを開始せず、しかも、適正なパイプラインのフローは比較的遅く処理されなければならない。

【0183】サイクル#2にて、ブランチグループAがデコードされ、サイクル#3にて、そのブランチグループAがプロセッサ203により実行される。上記規則3により、Aのサイクルのうちの実行にて、すなわち、サイクル#3にて、BがAの後継者になることが分かる。そのため、Bというアドレスはアドレスステージに置かれ、しかも、サイクル#3にてPPFAR702にロードされる。

【0184】MBHT701, 711が依然空であるので、BはAの場合と全く同一の方法でパイプラインを通過する。1つの例外は、Bがサイクル#6にて実行し、その後継者がCとなると、そのラン(run)の第1のトリプルを表すことである。すなわち、系列ABCが発見される。そのため、上記規則2により、トリプルABCがサイクル#7のENDOPで現れる。これが唯一のENDOPトリプルであるので、規則5と関係がなく、しかも、トリプルABCはサイクル#7にMBHT701, 711に送信される。

【0185】前のサイクルでアドレスCに対して探索がステージングされるので、1次MBHT701がサイクル#7にてビジーであることに注意すべきである。そのた

め、到達トリプルABCはサイクル#7にて1次更新待ち行列706に置かれる。1次探索はサイクル#7にてステージングされないで、1次MBHT701がサイクル#8にて更新に対して利用可能になる。出力カラムは"Update-ABC"とサイクル#8にてラベルが付され、更新アクティビティを示す。ABCは、更新が行われるとき、更新待ち行列から待機解除されことに注意すべきである。また、規則6により、サイクル#7にてミスがあるので、サイクル#7にて、PPFAR702がロードされた場合、その更新に探索より優先の優先順位が与えられることに注意すべきである。

【0186】2次MBHT711に対する探索が1次MBHT701のnビットに対してのみ生成され、しかも、1次ヒットがないので、2次MBHT711はサイクル#7にて遊休状態である。そのため、到達トリプルABCは直ちに2次MBHT711にサイクル#7にてストアされる。出力カラムはサイクル#7にて"Update-ABC"とラベルが付され、更新を示す。また、上述したように、トリプルABCはカッコで囲んで、同時に2次更新待ち行列116に待ち行列化されかつ待機解除されることを示す。このような表記を用いて、エントリABCはそのサイクルの終りに待ち行列にはないが、エントリがその待ち行列にとどまるべきか否かを判定するサイクルの間、その待ち行列はビジーであることを伝える。

【0187】ブランチグループを実行し、ENDOPトリプルを用いてMBHT701、711を更新することは、サイクル#15まで、同一の方法で継続される。サイクル#15では、ブランチグループA、B、C、D、およびEが全て実行され、トリプルABC、BCD、およびCDEが発見され、両テーブルにエンタされる。サイクル#15では、Eが実行されると、Eの後継者としてAが発見される。アドレスAがステージングされて命令取り出しが行われ、しかも、このサイクルで、PPFAR702にロードされる。

【0188】このとき、アドレスAを用いて1次MBHT701を探索することにより、サイクル#16でヒットし、しかも、その出力がそれぞれBおよびCになる。このヒットは上述したサイクル#8にてトリプルABCをストアした結果である。そのヒットにより、アドレスBおよびCがステージングされ、命令取り出しが行われ、アドレスCがPPFAR702にロードされ、アドレスBがサイクル#16にてSPFAR712にロードされる。また、そのヒットは、系列ABCが来る(upcoming)という予想が行われることを表す。そのため、トリプルABCが1次保留予想待ち行列704に置かれる。

【0189】サイクル#17にて、1次MBHT701がアドレスCを用いて探索された結果は、出力対DおよびEである。これらのアドレスはステージングされ、次の命令を取り出し、しかも、トリプルCDEが1次保留予想待ち行列704に待ち行列化される。2次MBHT7

11をアドレスBを用いて探索した結果は、出力対CおよびDである。第1出力CはPPFAR702の内容(Cである)と比較されるが、訂正アクションは行われない。PPFAR702にEがロードされ、しかも、PPFAR712にDがロードされる。

【0190】トリプルEABはまだ生じていないので、1次MBHT701がアドレスEにて探索した結果、サイクル#8にてミスが生じる。これは、1次更新待ち行列706のトリプルDEAが、規則6により、サイクル#19にて1次MBHT701にエンタされることを保証することにより、出力対EおよびAが生じる。1次ミスがあるので、2次テーブルの出力を用いて、規則3により、命令取り出しをドライブする。2次テーブルは、命令取り出しの前に、1つのブランチグループのみをステイ(stay)するので、単一の取り出しAのみがサイクル#18にてステージングされ、AはPPFAR702にロードされる。規則6により行われる更新に起因して、Aでの探索が1サイクルだけ遅延されるので、AはこのサイクルにてPPFAR712にカッコで囲んで示されることに注意すべきである。

【0191】サイクル#19にて、プロセッサ203はブランチグループの第1同時対、すなわち、BおよびCを実行する。1次MBHT711はトリプルDEAを用いて更新され、PPFAR702は引き続きアドレスAを保持する。2次MBHT711が遊休状態にあるので、2次MBHT711がこのサイクルでENDOPすると、更新するため直ちにトリプルFABを受け取る。1次MBHTがトリプルDEA更新中であるので、新しく到達したトリプルEABが更新待ち行列706、716に待ち行列化される。

【0192】サイクル#20にて、2つのトリプルABCおよびBCDは同時にENDOPする。第1トリプルABCは1次MBHT701に送信され、しかも、第2トリプルBCDが規則5により2次MBHT711に送信される。1次MBHT701にて、トリプルABCが保留予想待ち行列204で発見される。トリプルABCは1次MBHT701が既にトリプルの知識を有することを示す。その結果生じるアクションは、規則7により、ABCの1次保留予想待ち行列704での待機が解除される。同様に、2次MBHT711にて、トリプルBCDが保留予想待ち行列714にて発見される。トリプルBCDは2次MBHT711が既にトリプルの知識であることを示す。その結果生じるアクションは規則7によりBCDの2次保留予想待ち行列714での待機が解除される。

【0193】サイクル#21にて、2つのトリプルCDEおよびDEAは同時にENDOPする。サイクル#20でのアクションと同様のセットが生じ、その結果、CDEの保留予想待ち行列での待機が解除され、しかも、DE

Aの2次保留予想待ち行列711での待機が解除される。アドレスAがサイクル#21にてPPFAR702にロードされる。

【0194】サイクル#22にて、アドレスEでの1次MBHT701を探索した結果はミスである。ENDOPトリプルEABがサイクル#19にて生じ、しかも、更新待ち行列706に存在するが、1次MBHT701はサイクル#15から充分に利用され、しかも、EABの更新はまだ行われないことに注意すべきである。したがって、1次MBHT701はアドレスEでミスする。規則6は更新が次のサイクルで行われることを保証することに注意すべきである。これは、規則6を有するからか、あるいは規則6のようなものを有するからである。

【0195】サイクル#23にて、1次MBHT701がEABを用いて更新される。このサイクルから後は、5つのトリプルABC、BCD、CDE、DEA、およびEABが両テーブルにより示され、命令取り出しが定常状態に到達する。マルチブランチ予測装置700は次の11サイクルの間にサイクルごとに2つの取り出しを正常に開始し、しかも、プロセッサはサイクル#27-#34の間にサイクルごとに2つのブランチグループを実行する。

【0196】サイクル#34にて、ブランチグループの第7実行が生じ、しかも、今度は、後継者がXになるが、最初の6つの実行の場合のようにCではない。これは不正確なブランチ予測を表し、パイプラインがサイクル#35にてリスタートされる。

【0197】アドレスXのステージングを行い命令取り出しを行い、パイプラインの残りをクリアし、保留予想待ち行列704、714をクリアし、しかも、MBHT701、711の出力721、722、723、724を抑制することにより、サイクル#35のリスタートが行われる。この抑制は、APPENDIX Aの出力をカッコが囲むことにより表される。パイプラインは系列A、B、X、Y等の処理を開始し、ついで、2つのMBHT701、711が結局新しいトリプルを用いて更新される。

【0198】注目すべき点はサイクル#42である。このサイクルでは、1次MBHT701は、AがBおよびCの後にくることを忘れない。Cは最早実際の系列の一部ではないことに注意すべきである。しかしながら、命令取り出しはCのためにステージングされ、しかも、PPFAR702にCがロードされる。一方、SPFAR712にBがロードされる。

【0199】サイクル#43にて、2次MBHT711はBの後にXおよびYが来ることを示す。比較器718はダイバーゼンスを検知し、2次MBHT711出力723、724に従って命令取り出しをリダイレクトする同一のダイバーゼンスが再びサイクル#48にて検知されることに注意すべきである。これは、実際のトリプルABXがまずサイクル#35にてトリプル対の第2トリプルと

してENDOPされるからである。規則5による待ち行列の帯域幅制限により、このトリプルを1次MBHT701に送信することができない。トリプルABXはまず1次MBHT701にサイクル#46にて伝送されるが、1次MBHT701がビジーであるので、APPENDIXのこのトリプルを用いて更新されることはない。

【0200】1次MBHT701はサイクルごとに前もって2つのブランチグループを予測するので、しかも、系列A、B、X、およびYが4つのブランチグループを含むので、1次MBHT701は第2のリダイレクションの後にトリプルABXについて知る必要はない。サイクル#48から先、1次MBHT701がアドレスYおよびBでのみ探索されるが、このフローで、1次MBHT701が再びトリプルBAXに遭遇することはない。

【0201】サイクル#48から先は、再び、命令取り出しが、サイクル当たり2つのブランチグループの定常状態レートでヒットする。サイクル#51から先は、プロセッサ実行はサイクル当たり2つのブランチグループの定常状態レートでヒットする。以上でこの例の説明を終了する。

【0202】以上、本発明の実施例を説明したが、本発明の精神および範囲を逸脱することなく、種々の変更を行えることは当業者にとって当然である。

【0203】

【表1】

APPENDIX

表2	表3	表4
表5	表6	表7

【0204】

【表2】

【0205】
【表3】

クロック スーバスカラ プロセッサ 203							
サイクル	命令取り出し		プロセッサ		ENDOP トリプル		
	アドレス ステージ	キャッシュ ステージ	デコード	実行	トリプル 1	トリプル 2	
0	A	-	-	-	-	-	
1	-	A	-	-	-	-	
2	-	-	A	-	-	-	
3	B	-	-	A	-	-	
4	-	B	-	-	-	-	
5	-	-	B	-	-	-	
6	C	-	-	B	-	-	
7	-	C	-	-	ABC	-	
8	-	-	C	-	-	-	
9	D	-	-	C	-	-	
10	-	D	-	-	BCD	-	
11	-	-	D	-	-	-	
12	E	-	-	D	-	-	
13	-	E	-	-	CDE	-	
14	-	-	E	-	-	-	
15	A	-	-	E	-	-	
16	B, C	A	-	-	DEA	-	
17	D, E	B, C	A	-	-	-	
18	A	D, E	B, C	A	-	-	
19	-	A	D, E	B, C	EAB	-	
20	B, C	-	A	D, E	ABC	BCD	
21	D, E	B, C	-	A	CDE	DEA	
22	A	D, E	B, C	-	EAB	-	
23	-	A	D, E	B, C	-	-	
24	B, C	-	A	D, E	ABC	BCD	
25	D, E	B, C	-	A	CDE	DEA	

10

20

30

1次MBHT 701							比較器
出力			保留予想待ち行列 704		更新待ち行列 706		
PPFAR 702	PBG1 721	PBG2 722	待ち行列化	待機解除	待ち行列化	待機解除	
A	-	-	-	-	-	-	718
-	miss	miss	-	-	-	-	-
-	-	-	-	-	-	-	-
B	-	-	-	-	-	-	-
-	miss	miss	-	-	-	-	-
-	-	-	-	-	-	-	-
C	-	-	-	-	-	-	-
-	miss	miss	-	-	ABC	-	-
-	update-ABC	-	-	-	-	ABC	-
D	-	-	-	-	-	-	-
-	miss	miss	-	-	BCD	-	-
-	update-BCD	-	-	-	-	BCD	-
E	-	-	-	-	-	-	-
-	miss	miss	-	-	CDE	-	-
-	update-CDE	-	-	-	-	CDE	-
A	-	-	-	-	-	-	-
C	B	C	ABC	-	DEA	-	-
E	D	E	CDE	-	-	DEA	C=C
(A)	miss	miss	-	-	-	-	E=E
A	update-DEA	-	-	-	EAB	DEA	-
C	B	C	ABC	ABC	-	-	-
E	D	E	CDE	CDE	-	-	C=C
(A)	miss	miss	-	-	(EAB)	-	E=E
A	update-EAB	-	-	-	-	EAB	-
C	B	C	ABC	ABC	-	-	-
E	D	E	CDE	CDE	-	-	C=C

2次MBHT 711						
出力			保留予想待ち行列 714		更新待ち行列 716	
SPFAR 712	SBG1 723	SBG2 724	待ち行列化	待機解除	待ち行列化	待機解除
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	update-ABC	-	-	-	(ABC)	(ABC)
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	update-BCD	-	-	-	(BCD)	(BCD)
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	update-CDE	-	-	-	(CDE)	(CDE)
-	-	-	-	-	-	-
-	-	-	-	-	-	-
B	update-DEA	-	-	-	(DEA)	(DEA)
D	C D	BCD	-	-	-	-
-	E A	DEA	-	-	-	-
-	update-EAB	-	-	-	(EAB)	(EAB)
B	-	-	-	BCD	-	-
D	C D	BCD	DEA	-	-	-
-	E A	DEA	-	EAB	-	-
-	-	-	-	-	-	-
B	-	-	-	BCD	-	-
D	C D	BCD	DEA	-	-	-

【0208】

【表6】

26	A,B	D,E	B,C	-	EAB	-
27	C,D	A,B	D,E	B,C	-	-
28	E,A	C,D	A,B	D,E	ABC	BCD
29	B,C	E,A	C,D	A,B	CDE	DEA
30	D,E	B,C	E,A	C,D	EAB	ABC
31	A,B	D,E	B,C	E,A	BCD	CDE
32	C,D	A,B	D,E	B,C	DEA	EAB
33	E,A	C,D	A,B	D,E	ABC	BCD
34	B,C	E,A	C,D	A,B	CDE	DEA
35	X	clear	clear	clear	EAB	ABX
36	-	X	-	-	-	-
37	-	-	X	-	-	-
38	Y	-	-	X	-	-
39	-	Y	-	-	BXY	-
40	-	-	Y	-	-	-
41	A	-	-	Y	-	-
42	B,C	A	-	-	XYA	-
43	X,Y	B, (C)	A	-	-	-
44	-	X,Y	B	A	-	-
45	-	-	X,Y	B	YAB	-
46	A	-	-	X,Y	ABX	-
47	B,C	A	-	-	BXY	XYA
48	X,Y	B, (C)	A	-	-	-
49	A,B	X,Y	B	A	-	-
50	X,Y	A,B	X,Y	B	YAB	-
51	A,B	X,Y	A,B	X,Y	ABX	-
52	X,Y	A,B	X,Y	A,B	BXY	XYA
53	A,B	X,Y	A,B	X,Y	YAB	ABX
54	X,Y	A,B	X,Y	A,B	BXY	XYA
55	A,B	X,Y	A,B	X,Y	YAB	ABX

10

20

30

B	A	B	EAB	-	(EAB)	-	E=E
D	C	D	BCD	-	-	-	B=B
A	E	A	DEA	ABC	-	-	D=D
C	B	C	ABC	CDE	-	-	A=A
E	D	E	CDE	EAB	-	-	C=C
B	A	B	EAB	BCD	-	-	E=E
D	C	D	BCD	DEA	-	-	B=B
A	E	A	DEA	ABC	-	-	D=D
C	B	C	ABC	CDE	-	-	A=A
X	(D)	(E)	clear	clear	(EAB)	-	-
-	miss	miss	-	-	-	-	-
-	-	-	-	-	-	-	-
Y	-	-	-	-	-	-	-
-	miss	miss	-	-	BXY	-	-
-	update-BXY	-	-	-	-	BXY	-
A	-	-	-	-	-	-	-
C	B	C	ABC	-	XYA	-	-
Y	(D)	(E)	CDE	-	-	-	X≠C
-	miss	miss	-	-	-	-	-
-	update-XYA	-	-	-	YAB	XYA	-
A	update-YAB	clear	clear	ABX	YAB	-	-
C	B	C	ABC	-	BXY	-	-
Y	(D)	(E)	-	-	-	-	X≠C
B	A	B	UAB	-	-	-	-
Y	X	Y	BXY	YAB	-	-	B=B
Y	A	B	YAB	-	(ABX)	-	Y=Y
B	A	B	YAB	BXY	-	-	B=B
Y	X	Y	BXY	YAB	-	-	B=B
B	A	B	YAB	YAB	-	-	Y=Y
Y	X	Y	BXY	BXY	-	-	B=B

【0209】

【表7】

A	E	A	DEA	-	(EAB)	-
C	B	C	ABC	-	-	-
E	D	E	CDE	BCD	-	-
B	A	B	EAB	DEA	-	-
D	C	D	BCD	ABC	-	-
A	E	A	DEA	CDE	-	-
C	B	C	ABC	EAB	-	-
E	D	E	CDE	BCD	-	-
B	A	B	EAB	DEA	-	-
-	(C)	(D)	clear	clear	ABX	-
-	update-EAB	-	-	-	-	EAB
-	update-ABX	-	-	-	-	ABX
-	-	-	-	-	-	-
-	update-BXY	-	-	-	(BXY)	(BXY)
-	-	-	-	-	-	-
B	-	-	-	-	-	-
-	X	Y	BXY	-	XYA	-
-	update-XYA	-	-	-	-	XYA
-	update-YAB	-	-	-	(YAB)	(YAB)
-	update-ABX	-	-	-	(ABX)	(ABX)
B	update-XYA	-	-	-	(XYA)	(XYA)
-	X	Y	BXY	-	-	-
A	-	-	-	-	-	-
X	B	X	ABX	-	YAB	-
A	Y	A	XYA	ABX	-	-
A	B	X	ABX	XYA	-	-
X	B	X	ABX	ABX	-	-
A	Y	A	XYA	XYA	-	-
X	B	X	ABX	ABX	-	-

【0210】

【発明の効果】以上説明したように、本発明によれば、上記のように構成したので、同時予測を行うことができる。

【図面の簡単な説明】

【図1】スーパーカラープロセッサを示すブロック図である。

【図2】分岐グループの系列と、分岐グループトリプルを含むプログラムの一例を示す図である。

【図3】1重多重分岐予測装置を示すブロック図である。

【図4】n重多重分岐予測装置を示すブロック図である。

【図5】多重予測装置の論理フローを示すフローチャートである。

トである。

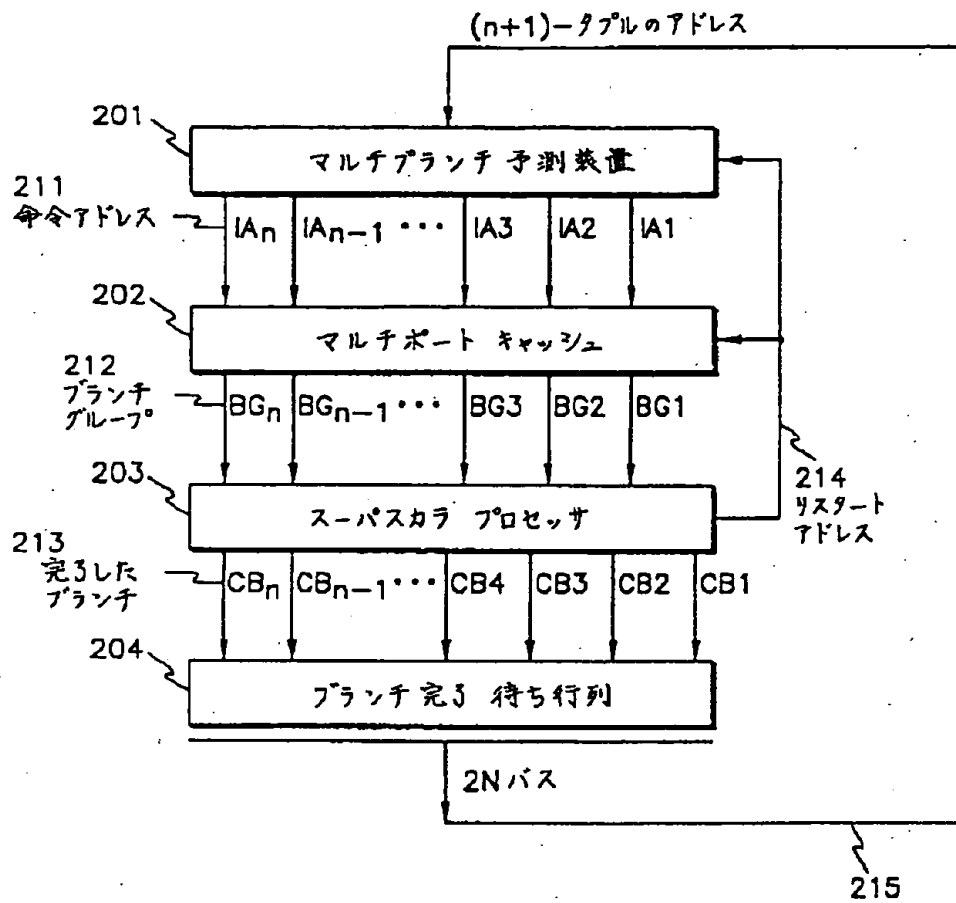
【図6】2重多重分岐予測装置を示すブロック図である。

【図7】4ステージパイプラインにおける仮想上の命令フローを示す図である。

【符号の説明】

- 201 マルチブランチ予測装置
- 202 マルチポートキャッシュ
- 203 スーパーカラープロセッサ
- 204 ブランチ完了待ち行列
- 211 命令アドレス
- 212 ブランチグループ
- 213 完了したブランチ
- 214 アドレスバス

【図1】



【図7】

	パイプライン #										
	1	2	3	4	5	6	7	8	9	10	11
DEC	I1	I2	I3	I4	BR			TARG	J1	J2	J3
AGEN		I1	I2	I3	I4	BR			TARG	J1	J2
CACHE			I1	I2	I3	I4	BR			TARG	J1
EXEC				I1	I2	I3	I4	BR			TARG

【図2】

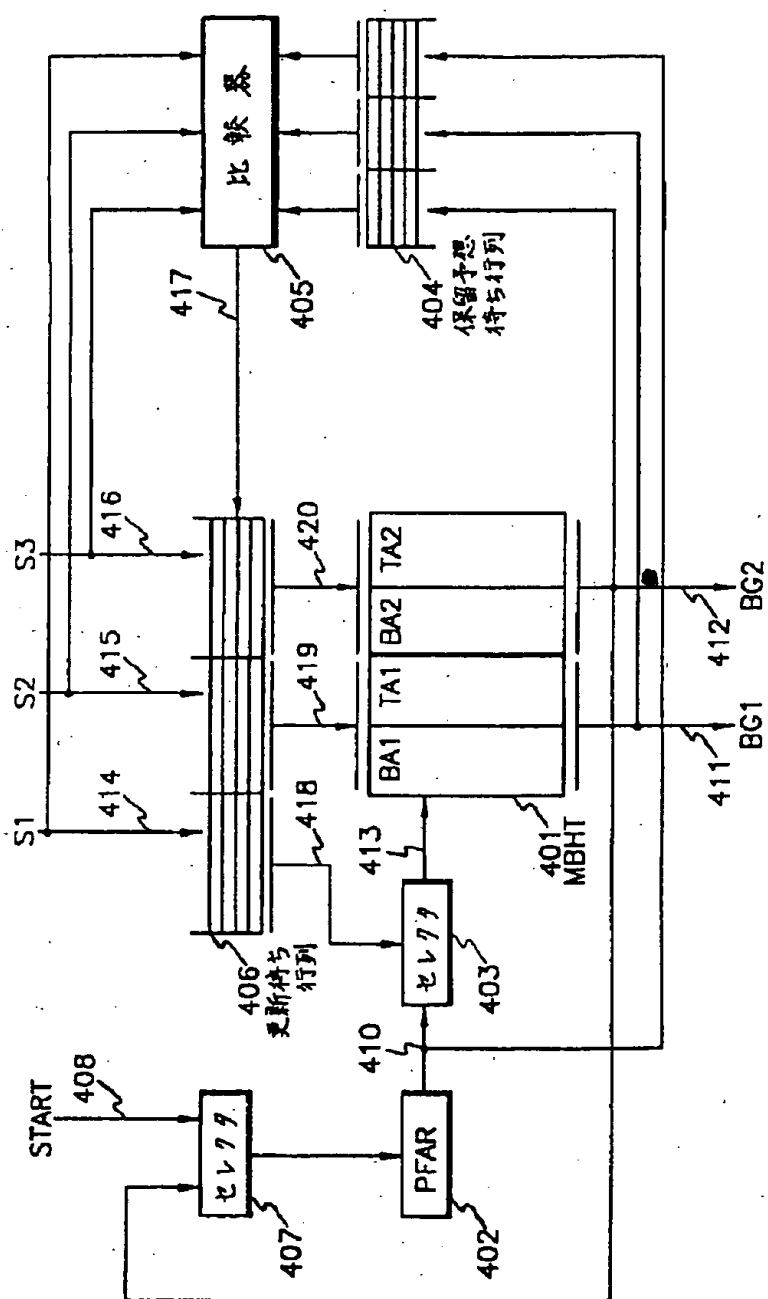
シンボリック プログラム

アドレス	命令
A	Instr A0
A+1	Instr A1
⋮	⋮
A+n _a -1	Go To B
⋮	⋮
D	Instr D0
D+1	Instr D1
⋮	⋮
D+n _d -1	Go To E
⋮	⋮
X	Instr X0
X+1	Instr X1
⋮	⋮
X+n _x -1	Go To Y
⋮	⋮
B	Instr B0
B+1	Instr B1
⋮	⋮
B+b	If <Cond> Go To C
⋮	⋮
B+n _b -1	Go To X
⋮	⋮
E	Instr E0
E+1	Instr E1
⋮	⋮
E+n _e -1	Go To A
⋮	⋮
Y	Instr Y0
Y+1	Instr Y1
⋮	⋮
Y+n _y -1	Go To A
⋮	⋮
C	Instr C0
C+1	Instr C1
⋮	⋮
C+n _c -1	Go To D

ブランチグループの
実行されたトリプル

A	B	C
B	C	D
C	D	E
D	E	A
E	A	B
A	B	C
B	C	D
C	D	E
D	E	A
E	A	B
A	B	C
B	C	D
C	D	E
D	E	A
E	A	B
A	B	X
B	X	Y
X	Y	A
Y	A	B
A	B	X
B	X	Y
X	Y	A
Y	A	B
A	B	C
B	C	D
C	D	E
D	E	A
E	A	B
A	B	C
B	C	D
⋮	⋮	⋮

【图 3】



【図4】

【図5】

【図6】

フロントページの続き

(72)発明者 ヨシュア ウィルソン ナイト
アメリカ合衆国 10547 ニューヨーク州
モヒーガン レイク サガモア アペニ
ユ 3490

(72)発明者 ジェイムズ ハーバート ボメレン
アメリカ合衆国 10514 ニューヨーク州
チャパックア ノース ベッドフォード
ロード 403

(72)発明者 トーマス アール. プザック
アメリカ合衆国 06877 コネチカット州
リッジフィールド スクールハウス プ
レイス 9